USERS

INCLUYE VERSIÓN DIGITAL GRATIS

PROTEUS VSM

SIMULACIÓN DE CIRCUITOS ELECTRÓNICOS

- DIBUJO DE DIAGRAMAS Y SIMULACIÓN INTERACTIVA CON ANIMACIÓN
- UTILIZACIÓN DE INSTRUMENTOS DE MEDICIÓN VIRTUAL
- GENERACIÓN DE SEÑALES ANÁLOGAS Y DIGITALES
- DEPURACIÓN DE PROGRAMAS PARA MICROCONTROLADORES
- CREACIÓN Y EDICIÓN DE COMPONENTES
- LOCALIZACIÓN Y CORRECCIÓN DE ERRORES DE SIMULACIÓN

por VÍCTOR ROSSANO

APRENDA A DISEÑAR CUALQUIER CIRCUITO DESDE SU PC



PROTEUS VSM

SIMULACIÓN DE CIRCUITOS ELECTRÓNICOS

por Víctor Rossano





TÍTULO: Proteus VSM
AUTOR: Víctor Rossano
COLECCIÓN: Manuales USERS
FORMATO: 17 x 24 cm
PÁGINAS: 352

ISBN 978-987-1857-72-2

Rossano, Víctor
Proteus VSM. - 1a ed. - Buenos Aires : Fox Andina, 2013.
352 p. ; 24x17 cm. (Manual users; 239)
ISBN 978-987-1857-72-2
1. Informática. I. Título
CDD 005.3

Víctor Rossano



Es ingeniero en Electrónica con especialidad en Sistemas Digitales. Cursó la carrera en la ESIME (Escuela Superior de Ingeniería Mecánica y Eléctrica), perteneciente al Instituto Politécnico Nacional, en la ciudad de México. Ha desarrollado su experiencia profesional en empresas dedicadas al ramo de la televisión, las telecomunicaciones y, de forma independiente, en computación. Desde siempre ha sido un apasionado por los temas relacionados con la ciencia y la tecnología.

Es autor de los libros *Electrónica Digital* y *Electrónica y microcontroladores PIC*, de esta misma editorial.

Prólogo



Cuando cursaba la carrera de Ingeniería en Electrónica, allá por mediados de los 90, las computadoras personales eran algo prácticamente inalcanzable para la mayoría debido a sus elevados costos. Era muy poco común que alguien contara con una computadora personal en casa. Las únicas máquinas que podíamos usar eran las de los laboratorios de computación de la escuela, que solo corrían el sistema operativo MS-DOS, y teníamos que esperar largo tiempo para tomar un turno de una hora. Así que prácticamente todo lo que estudiamos en la escuela fue siempre de forma empírica: aprendíamos sobre los componentes y circuitos en las clases o los libros, y luego los construíamos físicamente para ver si funcionaban según lo estudiado.

La primera computadora que pude tener en casa llegó cuando había terminado la carrera. Al tener acceso a Internet con el flamante módem de 56 k, comencé a buscar programas relacionados con la Electrónica. Fue entonces cuando me encontré con Proteus por primera vez, era la versión 4 o 5. Apenas comencé a utilizarlo, me gustó. Poco a poco fui aprendiendo más sobre el funcionamiento del programa, y con cada versión aparecían mejoras y nuevos componentes para simular en él. Desde ese tiempo lo he usado para simular todo tipo de circuitos electrónicos, y la gran mayoría de los simulados con éxito funcionaron correctamente en la realidad.

Me hubiera gustado tener herramientas como Proteus en mi tiempo de estudiante. Aunque no me quejo de haber tenido que armar todos mis circuitos físicamente en la escuela, ya que esto da experiencia y conocimiento, pero el hecho de contar con un simulador tan poderoso acelera el aprendizaje y facilita el diseño.

He intentado volcar en esta obra todo mi conocimiento y experiencia al utilizar Proteus desde hace ya un largo tiempo, para que el lector pueda convertirse en un verdadero experto en la simulación de circuitos electrónicos.

Víctor Rossano

El libro de un vistazo

Este libro está destinado a quienes tengan conocimientos de electrónica, al menos básicos, y quieran aprender a utilizar el software Proteus VSM para simular circuitos electrónicos. Veremos cómo dibujar los circuitos, y explicaremos en detalle todas las herramientas de simulación y análisis con las que cuenta el programa.



Para comenzar a simular circuitos electrónicos en Proteus, primero debemos aprender cómo está organizada su interfaz y sus funciones básicas. En este capítulo, conoceremos las bases del dibujo de diagramas de circuitos en el módulo ISIS, para después poder entrar en la simulación.



En este capítulo estudiaremos algunas opciones adicionales de dibujo de diagramas, como las terminales. Haremos un análisis de los componentes disponibles para las simulaciones y sentaremos las bases de la simulación de circuitos. Veremos las primeras herramientas de análisis, como las sondas de voltaje y de corriente.



Es fundamental saber cómo generar la alimentación de los circuitos para las simulaciones, tanto de voltajes de corriente directa, como de otros tipos de señales analógicas y digitales. En este capítulo veremos cómo se configuran las líneas de alimentación y de qué manera es posible obtener todo tipo de señales mediante los generadores.



Como sabemos, Proteus es un completo laboratorio virtual de electrónica, y cuenta con una gran cantidad de instrumentos de medición y análisis, que se asemejan mucho a los reales, como los osciloscopios, los voltímetros, los amperímetros y los analizadores lógicos. En este capítulo nos dedicaremos a estudiar en profundidad las características y el uso de cada una de estas herramientas.



Existe una forma de simulación no interactiva mediante la cual es posible realizar un análisis muy detallado de las señales que generan los circuitos simulados. En este capítulo estudiaremos cómo utilizar los gráficos de simulación para obtener una lectura de diversos parámetros en nuestros circuitos.



*06 SIMULACIÓN DE CIRCUITOS CON MICROCONTROLADORES PIC

Proteus es capaz de simular circuitos que incluyen microcontroladores, y contiene una amplia gama de ellos en sus librerías. En este capítulo aprenderemos a simular circuitos con microcontroladores PIC y cómo hacer la depuración de los programas que estos correrán.



*07 MANEJO AVANZADO DE DIAGRAMAS Y COMPONENTES

En este capítulo explicaremos cómo se pueden dividir los diagramas en varias hojas y cómo crear subcircuitos. Aprenderemos a editar los componentes para adaptarlos a nuestras necesidades o crear nuevos símbolos de componentes desde cero.



*08 PERSONALIZACIÓN DE LA INTERFAZ DE ISIS

La interfaz de ISIS puede ser personalizada cambiando de lugar algunas herramientas. También es posible modificar la forma en que vemos los circuitos en la interfaz variando los colores y los textos. En este capítulo aprenderemos a hacer todas estas modificaciones y, además, veremos cómo crear plantillas de estilos personalizadas.



*09 OPCIONES DE SIMULACIÓN Y CORRECCIÓN DE ERRORES

Algunas veces, las simulaciones pueden fallar debido a errores en el diseño de los circuitos. En este capítulo estudiaremos cómo funciona el núcleo del simulador, cómo configurar sus opciones, y daremos consejos y trucos para corregir los errores si estos se presentan al intentar simular un circuito.



SERVICIOS AL LECTOR

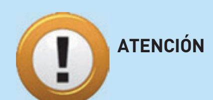
En este apartado final incluimos un índice temático y una completa guía de sitios web recomendados, donde encontraremos más información y recursos acerca del diseño y la simulación de circuitos electrónicos en Proteus.



INFORMACIÓN COMPLEMENTARIA

A lo largo de este manual podrá encontrar una serie de recuadros que le brindarán información complementaria: curiosidades, trucos, ideas y consejos sobre los temas tratados. Para que pueda distinguirlos en forma más sencilla, cada recuadro está identificado con diferentes iconos:

 CURIOSIDADES E IDEAS

 ATENCIÓN

 DATOS ÚTILES Y NOVEDADES

 SITIOS WEB

Contenido

*03

Líneas de alimentación y generadores

*04

Instrumentos de medición virtuales

*05

Análisis con gráficos



Simulación con microcontroladores PIC



Manejo de diagramas y componentes

*08

Personalización de la interfaz de ISIS

*09

Opciones de simulación y corrección de errores

*

Servicios al lector

Introducción



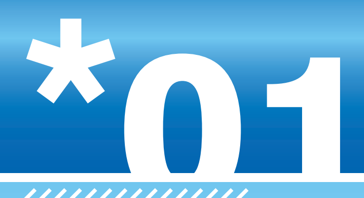
Proteus es considerado uno de los mejores y más completos programas para el diseño de circuitos electrónicos en la actualidad, no solo por su capacidad de simulación y análisis, sino también por el hecho de poder utilizar una gran cantidad de microcontroladores de diferentes familias. Esta obra está dirigida a todos aquellos que tienen conocimientos de electrónica y desean aprender el uso de este poderoso software.

Comenzaremos describiendo la interfaz de ISIS, que es el módulo donde se realiza la simulación, y aprenderemos a dibujar diagramas en él. Esta es la base para poder simular cualquier circuito, porque dibujar correctamente un diagrama facilita y acelera el proceso. Estudiaremos gran cantidad de opciones para hacerlo de forma eficiente, como, por ejemplo, la división de los circuitos en varias hojas de trabajo y el uso de subcircuitos.

Continuaremos conociendo las diferentes maneras de simulación en Proteus. Por un lado, la simulación interactiva en tiempo real, que permite observar el funcionamiento del circuito en el propio diagrama con múltiples opciones de animación, como si lo hiciéramos en la realidad mediante el uso de componentes interactivos: pulsadores, displays, teclados, interruptores, LEDs, sensores, etcétera. Por otro lado, la simulación no interactiva, en donde se pueden estudiar las señales generadas por el circuito o parte de él. Aprenderemos a analizar todo tipo de señales en los circuitos a través de múltiples herramientas de prueba y medición. Estudiaremos la simulación de circuitos con microcontroladores y cómo realizar la depuración de sus programas en Proteus.

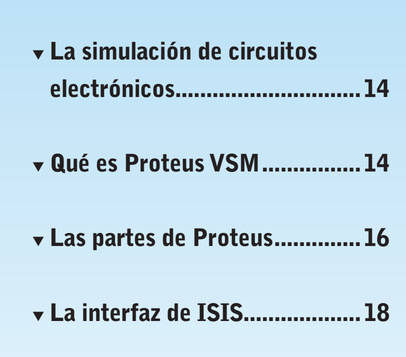
Para finalizar, veremos la manera de personalizar la interfaz del programa, y cómo corregir diferentes errores cuando estos se presenten y no permitan simular un circuito.

Todo esto lo haremos tomando ejemplos y proyectos reales para observarlos y aprender de forma práctica el uso del simulador. Los archivos de los proyectos o ejemplos mencionados a lo largo de la obra se pueden descargar desde el sitio web **www.redusers.com**.



Introducción a Proteus

En este capítulo conoceremos el programa Proteus VSM, su interfaz y sus funciones principales. Aprenderemos a elegir y manejar componentes, y comenzaremos a dibujar diagramas de circuitos electrónicos en el módulo ISIS. Para poder simular un circuito es necesario primero dibujarlo; con este objetivo, descubriremos las principales herramientas que nos permitirán hacerlo de manera fácil y rápida.







La simulación de circuitos electrónicos

En la actualidad, existen cada vez más programas dedicados al diseño asistido por computadora o **CAD** (**Computer-Aided Design**) para diferentes áreas de desarrollo, y la **electrónica** no podía quedarse atrás. Dentro de este contexto, en los últimos años han surgido gran cantidad de programas orien tados a los expertos en electrónica para ayudarlos en el diseño de circuitos.

Los programas tipo CAD enfocados en la electrónica en general tienen las siguientes características: dibujo de diagramas de circuitos, simulación de circuitos electrónicos y diseño de circuitos impresos (PCB). Algunos poseen solo una de estas características, pero otros pueden realizar las tres en un único paquete.



Qué es Proteus VSM

Proteus VSM es un sistema de diseño electrónico basado en la **simulación** analógica, digital o mixta de circuitos, que brinda la posibilidad de interacción con muchos de los elementos que integran el circuito. Incluye componentes animados para la visualización de su comportamiento en tiempo real, además de un completo sistema de generación y análisis de señales. También cuenta con un módulo para el diseño de circuitos impresos.

Las siglas **VSM** significan **Virtual System Modelling**, que en español podemos traducir como **sistema de modelado virtual**, ya que Proteus VSM permite modelar de forma virtual en la computadora prácticamente cualquier circuito.

La característica principal que hace de Proteus VSM uno de los simuladores preferidos por muchos aficionados y profesionales de la electrónica es la posibilidad de simular circuitos que incluyen microprocesadores o microcontroladores.

Aunque el nombre completo del programa es Proteus VSM, a partir de ahora nos referiremos a él solo como **Proteus**, para mayor facilidad.

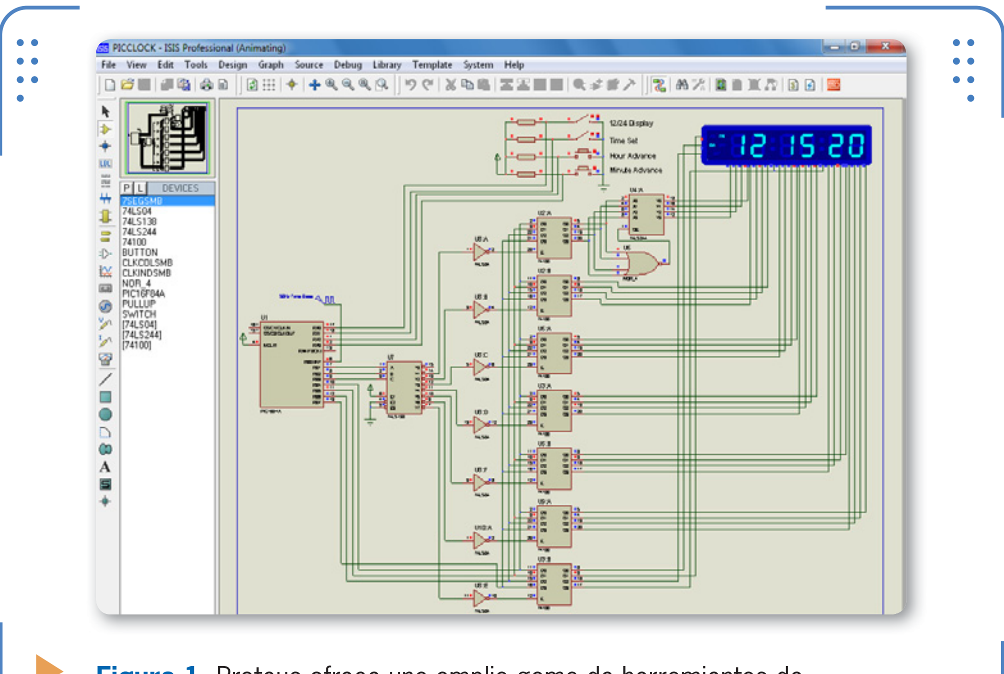


Figura 1. Proteus ofrece una amplia gama de herramientas de simulación en un entorno gráfico amigable y fácil de usar.

Si visitamos la página web **www.labcenter.com** de **Labcenter Electronics**, que es el desarrollador de Proteus, podremos descargar una versión demo del programa y, además, encontrar información acerca de las licencias, características, funciones, controladores y módulos extra, entre otros elementos.

Esta versión demo es completamente funcional, excepto por las siguientes limitaciones: no permite guardar nuestros circuitos si hemos creado uno desde cero, o si modificamos uno preexistente, no podremos almacenar los cambios realizados en él. Además, no es posible simular circuitos que incluyan microcontroladores.

Para instalar Proteus en nuestra computadora, simplemente debemos ejecutar el archivo de instalación y seguir los pasos tal como lo hacemos con cualquier otra aplicación en Windows. A partir de ahora, tomaremos una **versión completa** de Proteus para dar las explicaciones y ejemplos, es decir, una versión con licencia para usar todas sus funciones. Trabajaremos con la **versión 7.10**, que es la más reciente hasta el momento de la redacción de esta obra.

Las partes de Proteus

El programa cuenta con dos partes o componentes principales. Uno de ellos es el módulo **ISIS**, que es donde vamos a dibujar los diagramas de los circuitos electrónicos y, también, desde donde efectuaremos las simulaciones. Si es la primera vez que abrimos el módulo ISIS, después de instalar Proteus en el sistema, es posible que aparezca una ventana llamada **View Sample Designs**, que nos preguntará si queremos ver los diseños de ejemplo que se instalan junto con el programa.

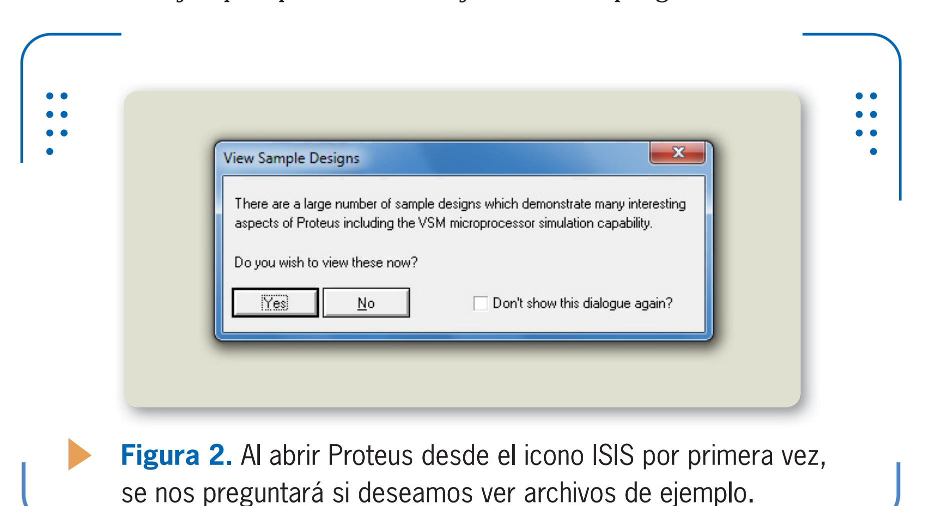


Figura 2. Al abrir Proteus desde el icono ISIS por primera vez, se nos preguntará si deseamos ver archivos de ejemplo.

Para ver los ejemplos, presionamos el botón **Yes**; de lo contrario, pulsamos **No**. Si marcamos la casilla **Don't show this dialog again?**, la ventana no volverá a aparecer; pero si después queremos acceder a los archivos de ejemplo, podremos hacerlo desde el menú **Help/Sample Designs**.



NO ESTUDIAREMOS ARES

Veremos que Proteus cuenta con dos módulos principales: ISIS y ARES. En ARES se diseñan placas de circuito impreso o PCB (**Printed Circuit Board**). En este manual solo nos vamos a referir al módulo ISIS y a la simulación de circuitos en él. No nos dedicaremos a estudiar el módulo ARES, debido a que la extensión de la obra no nos permite hacerlo en profundidad.

Además del módulo ISIS, Proteus cuenta con otro módulo, denominado **ARES**, que es donde se diseñan las placas de **circuito impreso** (PCB) utilizadas para el posterior armado de los circuitos. En relación a este tema, otra característica muy interesante de este programa es la posibilidad de importar un circuito dibujado en ISIS directamente al módulo ARES. De esta manera, podremos diseñar fácil y rápidamente un circuito impreso para él.

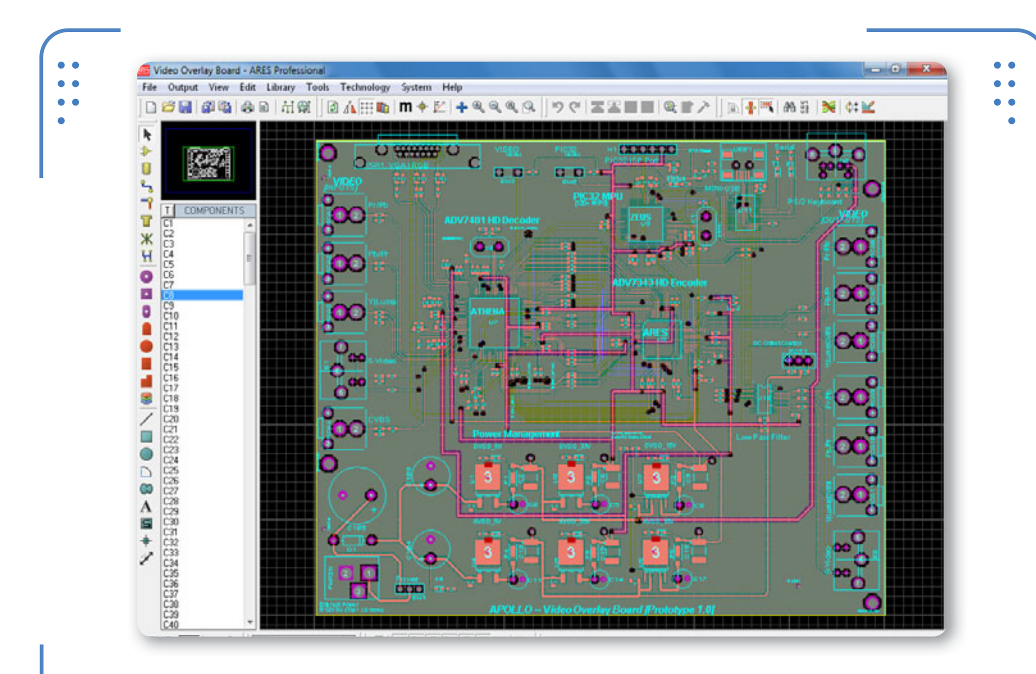


Figura 3. El módulo ARES permite el diseño de circuitos impresos donde construiremos los circuitos de forma física.



EL BORDE DE LA HOJA DE TRABAJO

Dentro de la interfaz de ISIS, veremos una línea de color azul cuya función es delimitar la hoja de trabajo, pero esta solo aparece a modo de referencia. En caso de que imprimamos el circuito o lo guardemos como una imagen, esta línea no se imprimirá ni se mostrará en la imagen almacenada. Si por alguna razón necesitamos que aparezca, tendremos que colocar un recuadro con elementos gráficos. Más adelante, en este mismo capítulo, veremos cómo podemos hacerlo.

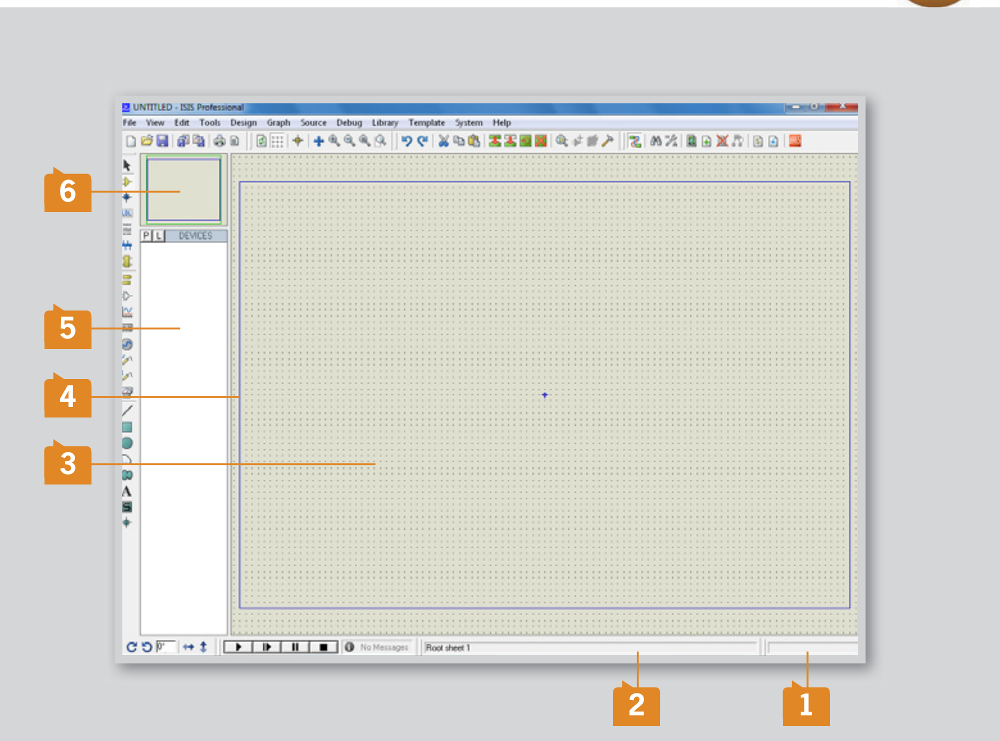


La interfaz de ISIS

En principio, el módulo ISIS es un espacio para dibujar los diagramas de nuestros circuitos. Al abrir Proteus desde el icono ISIS, veremos la pantalla que se explica en la siguiente **Guía visual**.

▼ GUÍA VISUAL ■ LA INTERFAZ DE ISIS





01 **BARRA DE COORDENADAS:** en esta barra podemos ver las coordenadas donde se encuentra el cursor en todo momento, mientras lo desplazamos por la ventana de edición.

02 **BARRA DE ESTADO:** aquí aparece información útil de lo que tengamos seleccionado o del elemento sobre el cual se encuentre el cursor del mouse en ese momento.

03 **VENTANA DE EDICIÓN:** es la ventana donde dibujaremos los circuitos electrónicos colocando los componentes, interconectándolos, y agregando también otros instrumentos y herramientas. ▶

04 **BORDE DE HOJA:** la línea de color azul determina el límite de la hoja de trabajo; debemos colocar el circuito dentro de ella.

05 **SELECTOR DE OBJETOS:** en esta ventana aparecerán los componentes y otros elementos disponibles, dependiendo del modo seleccionado, y desde allí podremos elegirlos para colocarlos en el diseño.

06 **VENTANA DE VISTA PREVIA:** esta pequeña ventana nos ofrece una vista previa del circuito o de los elementos que vayamos a colocar en el diseño; más adelante veremos detalles sobre su uso.

La hoja de trabajo

En la ventana de edición tenemos dos elementos principales: la **hoja de trabajo** y **la rejilla**. La hoja de trabajo está delimitada por el cuadro de color azul. Cuando accedemos a ISIS desde su icono, es decir, sin abrir ningún circuito, por defecto la ventana de edición está vacía, y el tamaño de la hoja de trabajo es de 10 por 7 pulgadas.

Si necesitamos cambiar el tamaño de la hoja de trabajo, podemos hacerlo mediante el menú **System/Set Sheet Sizes...**. Al seleccionar esta opción, aparece una ventana con el título **Sheet Size Configuration**, donde podemos elegir un tamaño de la lista o uno personalizado, llamado **User**. En realidad, es posible seleccionar un tamaño de la lista y luego ajustarlo a la medida que deseamos.

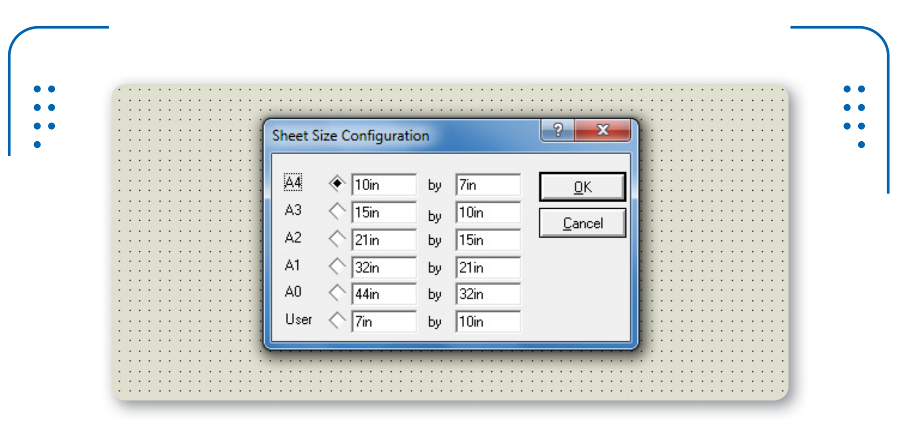


Figura 4. Sheet Size Configuration ofrece cinco tamaños de hoja, desde A4 a A0, y un tamaño definido por el usuario.

También podemos utilizar una plantilla diferente para la hoja de trabajo; si vamos al menú **File/New Design...**, aparecerá una lista de las plantillas que se instalan con Proteus.

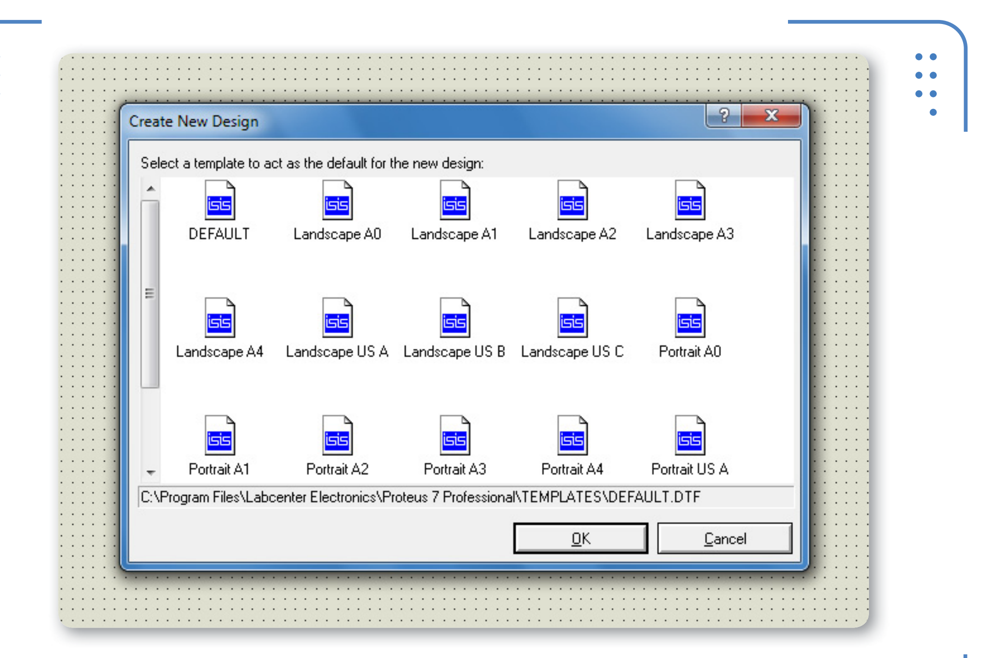


Figura 5. Podemos elegir una plantilla de la lista, incluyendo la que viene por defecto (**DEFAULT**).

Las plantillas adicionales contienen un **marco** con coordenadas y un **cuadro de datos** en la parte inferior derecha, que incluye la fecha, la hora y el nombre del archivo, entre otros datos. En el **Capítulo 8** veremos cómo manejar este cuadro de datos.



PLANTILLAS: BORDE Y CUADRO DE DATOS

Si en alguno de nuestros diseños utilizamos una plantilla de las que tienen coordenadas en el borde de la hoja y un cuadro de datos, debemos tener en cuenta que estos cuadros están construidos a partir de diferentes elementos gráficos y texto y, entonces, en esos casos, notaremos que sí se imprimirán o mostrarán en las imágenes que almacenemos.

Figura 6. Un nuevo documento vacío en ISIS, esta vez, con la plantilla llamada **Landscape A4**.

La rejilla

Además de la hoja de trabajo, en la ventana de edición podemos ver una **rejilla de puntos o líneas**, que nos servirá como guía al momento de dibujar los circuitos; podemos desactivarla u ocultarla si lo deseamos. Los puntos o líneas de la rejilla tienen, por defecto, una separación fija entre sí. Entre dos puntos o líneas hay una distancia de un décimo de pulgada; es decir, cada 10 puntos o líneas tendremos una pulgada.

SOLO VERSIÓN EN INGLÉS

Proteus es un programa desarrollado por una compañía inglesa, y hasta el momento solo tiene versión en inglés, lo cual puede resultar un pequeño obstáculo para quienes no dominan ese idioma. Es por eso que este manual seguramente será de gran utilidad para entender las características y las funciones de Proteus sin necesidad de conocer el idioma inglés.

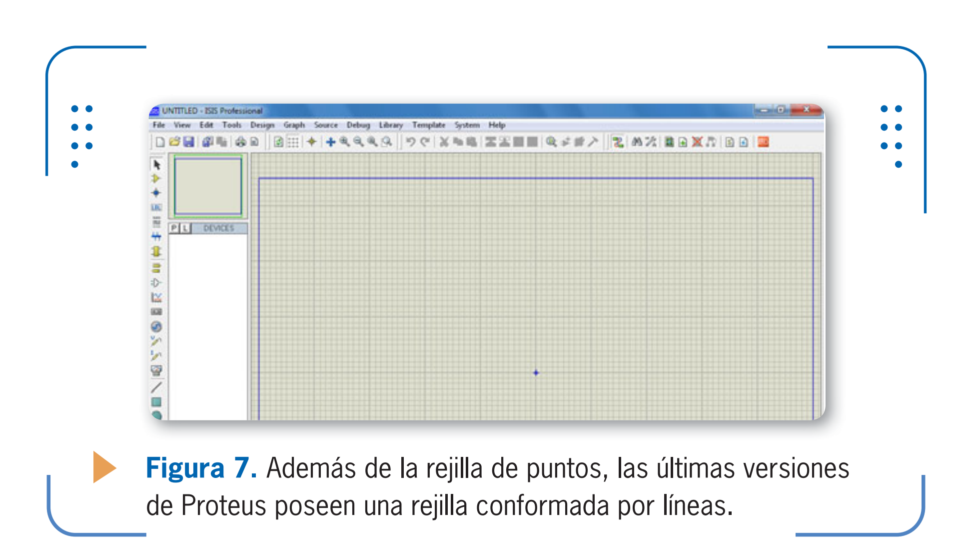


Figura 7. Además de la rejilla de puntos, las últimas versiones de Proteus poseen una rejilla conformada por líneas.

Es posible modificar la separación de los puntos o las líneas que conforman la rejilla, para lo cual tenemos que ir al menú **View** y elegir una de las siguientes opciones disponibles: **Snap 10th**, **Snap 50th**, **Snap 0.1in** o **Snap 0.5in**, que cambian la división de cada pulgada en 100, 50, 10 o 2, respectivamente. A la derecha de cada opción en el menú **View**, encontramos la indicación de las teclas que podemos usar para la misma función, es decir, **el atajo de teclado**, no solo para estas opciones sino para muchas de las de los menús. Una vez que incorporemos su uso, ahorraremos tiempo de trabajo.

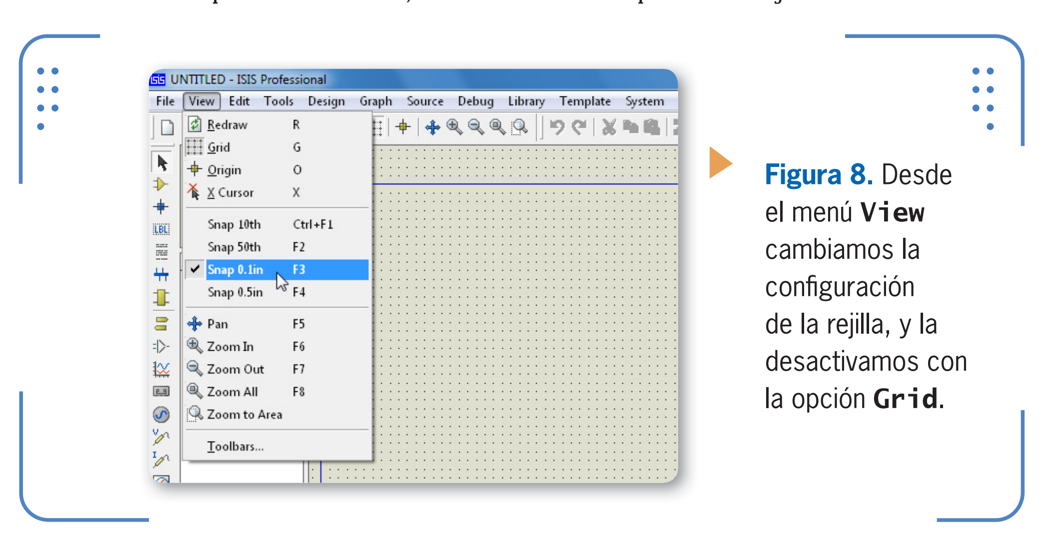


Figura 8. Desde el menú **View** cambiamos la configuración de la rejilla, y la desactivamos con la opción **Grid**.



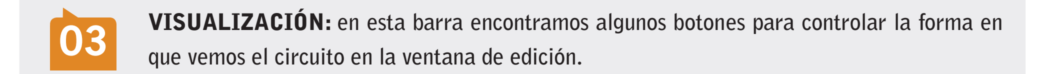
Las barras de herramientas

En la interfaz de Proteus, podemos encontrar diferentes **barras de herramientas**, que contienen botones para múltiples funciones del programa. En la siguiente **Guía visual** conoceremos cuáles son y dónde están ubicadas, por defecto, en la ventana de ISIS.

GUÍA VISUAL ■ BARRAS DE HERRAMIENTAS

01 **MODO:** esta barra de herramientas es la más importante porque nos permite elegir el modo en el que vamos a trabajar mientras dibujamos un circuito en la ventana de edición. Nos permitirá elegir y colocar componentes y otros elementos, como instrumentos de medición, gráficos de simulación, etcétera.

02 **OPCIONES DE ARCHIVO:** esta barra es muy similar a la de otros programas de Windows, y se utiliza para realizar operaciones básicas con los archivos, como crear uno nuevo, abrir uno existente, guardar el archivo en el que estamos trabajando, etcétera.



03 **VISUALIZACIÓN:** en esta barra encontramos algunos botones para controlar la forma en que vemos el circuito en la ventana de edición.

04 **EDICIÓN:** aquí tenemos varias opciones de edición de los diferentes elementos.

05 **HERRAMIENTAS DE DISEÑO:** esta barra muestra diversas opciones de diseño para trabajar en varias hojas, generar reportes, etcétera.

06 **SIMULACIÓN:** aquí podemos encontrar los controles de simulación. Hablaremos de esta barra en el **Capítulo 2**, donde comenzaremos a estudiar la simulación en Proteus.

07 **ROTACIÓN Y REFLEXIÓN:** esta barra, ubicada en la parte inferior izquierda de la pantalla, nos permite rotar y reflejar los componentes o elementos antes de colocarlos en un diseño.

Las herramientas de Modo

Veamos con más detalle la barra de herramientas de **Modo**. Esta se encuentra dividida en tres partes: la superior presenta las herramientas principales, en la parte media tenemos herramientas de dispositivos, y en la inferior, herramientas de dibujo en 2D.

LAS HERRAMIENTAS DE DIBUJO EN 2D PUEDEN USARSE PARA DIBUJAR COMPONENTES

Detallaremos aquí solo las herramientas principales de la barra de herramientas de **Modo** para poder comenzar a dibujar nuestros diagramas de circuitos; estudiaremos las demás herramientas en capítulos posteriores.

Debemos notar que al cambiar de modo (es decir, al presionar algunos de los botones de la barra de herramientas de **Modo**), la parte superior del **Selector de objetos** cambia de título, y pueden aparecer uno o dos pequeños botones.

Por otro lado, las herramientas de dibujo en 2D permiten colocar elementos gráficos que no son componentes en la ventana de edición, para crear, por ejemplo, líneas, marcos, cuadros de texto, flechas, textos indicativos, títulos, etcétera. También pueden usarse para dibujar nuevos componentes; en el **Capítulo 7** veremos en detalle el procedimiento para hacerlo.

GUÍA VISUAL ■ HERRAMIENTAS DE MODO



01 **MODO DE SELECCIÓN (SELECTION MODE):** con esta herramienta entramos en el modo de selección, es decir, seleccionamos cualquier objeto en la ventana de edición, haciendo un clic con el mouse sobre él, incluyendo componentes, herramientas o líneas de conexión.

02 **MODO COMPONENTE (COMPONENT MODE):** en este modo podemos seleccionar los componentes que formarán el circuito y, además, colocarlos en la ventana de edición.

03 **MODO DE PUNTOS DE UNIÓN (JUNCTION DOT MODE):** este modo permite colocar puntos de unión donde dos o más líneas de conexión se unen, aunque realmente no es muy usado, ya que los puntos de unión aparecen automáticamente al conectar dos líneas.

04 **MODO DE NOTAS DE TEXTO (TEXT SCRIPT MODE):** este modo brinda la posibilidad de colocar notas de texto en cualquier lugar del diseño, si lo necesitamos.

05 **MODO DE TERMINALES (TERMINALS MODE):** este modo permite colocar terminales en un diseño. Al presionar este botón, aparecerá la lista de terminales disponibles en la ventana **Selector de objetos**.

06 **MODO DE LÍNEA (2D GRAPHICS LINE MODE):** en este modo estamos trabajando con elementos gráficos; permite dibujar una línea recta en cualquier lugar del diseño.

07 **MODO DE CUADRADO (2D GRAPHICS BOX MODE):** este botón se utiliza para dibujar un cuadrado.

08 **MODO DE CÍRCULO (2D GRAPHICS CIRCLE MODE):** este botón se usa para dibujar un círculo.

09 **MODO DE ARCO (2D GRAPHICS ARCH MODE):** este botón permite dibujar un arco de circunferencia.

10 **MODO DE FORMA MIXTA (2D GRAPHICS CLOSED PATH MODE):** con este botón creamos una forma cerrada mixta, compuesta de líneas rectas o curvas. Al hacer clic con esta herramienta en la ventana de edición, podemos ir dibujando los segmentos de la figura hasta completarla, uniendo el último punto con el primero. Para trazar un segmento curvo, presionamos la tecla **CTRL**.

11 **MODO DE TEXTO (2D GRAPHICS TEXT MODE):** con este botón podemos colocar una línea de texto en nuestro diseño.

12 **MODO DE SÍMBOLOS (2D GRAPHICS SYMBOLS MODE):** este botón permite colocar algunos símbolos prediseñados en el diseño, los cuales se eligen de una librería.

13 **MODO DE MARCADORES (2D GRAPHICS MARKERS MODE):** similar al modo de símbolos, pero en este caso podemos colocar marcadores o elementos especiales de la lista que aparece al presionarlo.

Elegir componentes de las librerías

Veamos cómo elegir los componentes que formarán nuestros circuitos electrónicos en Proteus. Para seleccionar los componentes, debemos estar en el modo **Componente**, para lo cual, simplemente, presionaremos el botón **Component Mode** en la barra de herramientas de **Modo**.

Aparentemente no sucede nada al hacerlo, pero si observamos la ventana del **Selector de objetos**, en la parte superior aparecen dos pequeños botones, uno con una **P** y otro con una **L**. Además, a la derecha de ellos, en el título de la ventana del **Selector de objetos**, podemos apreciar la palabra **DEVICES**, que significa **dispositivos**.

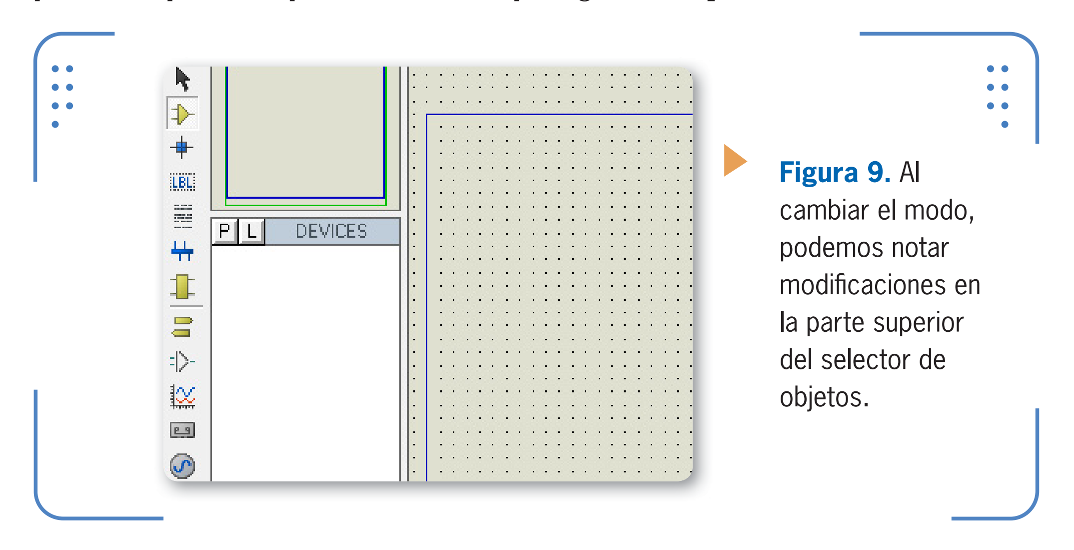


Figura 9. Al cambiar el modo, podemos notar modificaciones en la parte superior del selector de objetos.

Si presionamos el botón **P**, se abrirá la ventana **Pick Devices**, donde podemos navegar por las **librerías de componentes** para elegir los que formarán nuestro diseño. Desde esta ventana también es posible hacer una búsqueda concreta. En la siguiente **Guía visual** conoceremos cada uno de sus elementos.

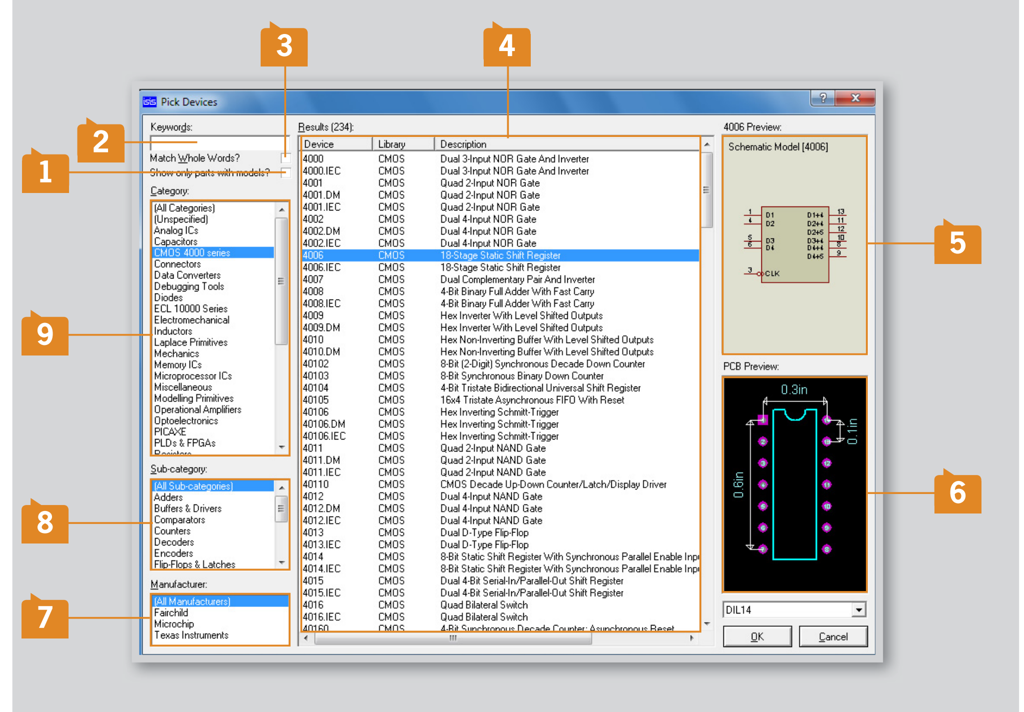


UBICACIÓN DE LAS BARRAS

Las barras de herramientas pueden cambiarse de lugar, arrastrándolas desde la pequeña barra a la izquierda o arriba de cada una, para ubicarlas en un sitio diferente si así lo deseamos. En esta obra todas las barras de herramientas se presentan en el lugar por defecto, no han sido movidas.

▼ GUÍA VISUAL ■ LA VENTANA PICK DEVICES





01 **SHOW ONLY PARTS WITH MODELS? (MOSTRAR SOLO COMPONENTES CON MODELO):** al seleccionar esta opción, únicamente se mostrarán en los resultados los componentes que cuenten con modelo, es decir, los que pueden simularse. Si la desmarcamos, se mostrarán todos sin distinción.

02 **KEYWORDS (CUADRO DE BÚSQUEDA):** mediante este campo realizamos búsquedas en las librerías de componentes. La búsqueda es sensible, es decir que a medida que vamos escribiendo en este cuadro, irán apareciendo las coincidencias en la ventana de resultados, por lo que muchas veces no será necesario ingresar todo el nombre del componente.

03 **MATCH WHOLE WORDS? (SOLO PALABRAS COMPLETAS):** si marcamos esta opción, la búsqueda se realizará solo con las palabras exactas que ingresemos en el cuadro.

04 **RESULTS (RESULTADOS):** en esta ventana aparecerá la lista con los resultados de la búsqueda o la navegación por las categorías. Estará dividida, básicamente, en: nombre de los componentes (**Device**), la librería a la que pertenece cada uno (**Library**) y una descripción (**Description**); puede haber otras columnas dependiendo de la categoría elegida. ▶

05 **PCB PREVIEW (VISTA PREVIA DE EMPAQUE):** en esta ventana podemos observar una vista previa del empaque del componente. Esto es útil si vamos a diseñar una placa de circuito impreso (PCB) para nuestro circuito usando el módulo ARES de Proteus. Debajo de esta ventana tenemos una lista para elegir el empaque, ya que muchos componentes pueden contar con más de uno.

06 **SCHEMATIC PREVIEW (VISTA PREVIA DE SÍMBOLO):** aquí tendremos una vista previa del símbolo del componente seleccionado en la ventana de resultados.

07 **MANUFACTURER (FABRICANTE):** en el caso de componentes específicos, en esta ventana podemos elegir al fabricante para filtrar aún más la navegación por la lista.

08 **SUB-CATEGORY (SUBCATEGORÍA):** muestra las subcategorías. Por ejemplo, si seleccionamos **Transistors** (transistores), en la subcategoría aparecerán los diferentes tipos de transistores, como bipolares, JFET, MOSFET, etcétera.

09 **CATEGORY (CATEGORÍA):** presenta las categorías en las que están divididos todos los componentes.

Al navegar o realizar una búsqueda en la ventana **Pick Devices**, es posible encontrar los componentes que necesitamos para dibujar los circuitos. Para elegir uno de ellos, podemos manejarnos de dos maneras.

Una es hacer doble clic sobre el componente deseado en la lista de resultados; esto hará que el nombre del componente se muestre en la ventana del **Selector de objetos**. De esta forma, podemos seguir navegando o buscando todos los componentes necesarios y agregándolos a la ventana del Selector. Cuando finalizamos, cerramos la ventana **Pick Devices** presionando el botón **Cancel** o **Cerrar**.

TENEMOS DOS MANERAS DE ELEGIR LOS COMPONENTES EN LA VENTANA PICK DEVICES



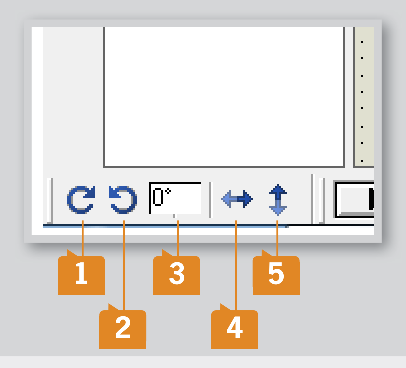
Otra posibilidad es hacer un clic sobre el componente para seleccionarlo de la lista, y después presionar **OK** en la parte inferior derecha de la ventana **Pick Devices**. De esta manera, el componente pasará al **Selector de objetos**, y la ventana **Pick Devices** se cerrará. Todo dependerá de si necesitamos usar solo un componente o más.

Las herramientas de Rotación y reflexión

Después de elegir los componentes en la ventana **Pick Devices** y tenerlos listos en la ventana del **Selector de objetos**, ya podemos colocarlos en el diseño. Antes de hacerlo, es conveniente analizar las opciones para rotarlos o reflejarlos según lo necesitemos. Para esto usaremos la barra de herramientas de **Rotación y reflexión**, que conoceremos en la siguiente **Guía visual**.

GUÍA VISUAL ■ ROTACIÓN Y REFLEXIÓN





01 **ROTAR EN SENTIDO DE LAS MANECILLAS DEL RELOJ:** este botón permite rotar el componente seleccionado en la ventana del **Selector de objetos** siguiendo el sentido de las agujas del reloj. Cada vez que lo presionamos, el componente se rotará 90 grados.

02 **ROTAR EN SENTIDO CONTRARIO DE LAS MANECILLAS DEL RELOJ:** permite rotar el componente seleccionado en la ventana del **Selector de objetos** en sentido contrario al de las agujas del reloj. Cada vez que lo presionamos, el componente se rotará 90 grados.

03 **ÁNGULO DE ROTACIÓN:** en este cuadro podemos colocar directamente el ángulo de rotación desde el teclado. Este valor debe ser múltiplo de 90 grados; es decir, 0, 90, -90, 180, -180, 270, -270.

04 **REFLEJAR EN EL EJE X:** refleja los componentes en el eje X.

05 **REFLEJAR EN EL EJE Y:** refleja el componente seleccionado en el eje Y.

Las herramientas de la barra de **Rotación y reflexión** nos permiten cambiar la orientación de los componentes antes de colocarlos en el diseño. Es importante saber que estos botones no funcionan con los componentes ya colocados en la ventana de edición.

Al seleccionar algún componente en la ventana del **Selector de objetos**, este aparecerá en la ventana de **Vista previa**, donde observaremos su símbolo y, además, al rotar o reflejar, también veremos cuál es la nueva orientación.

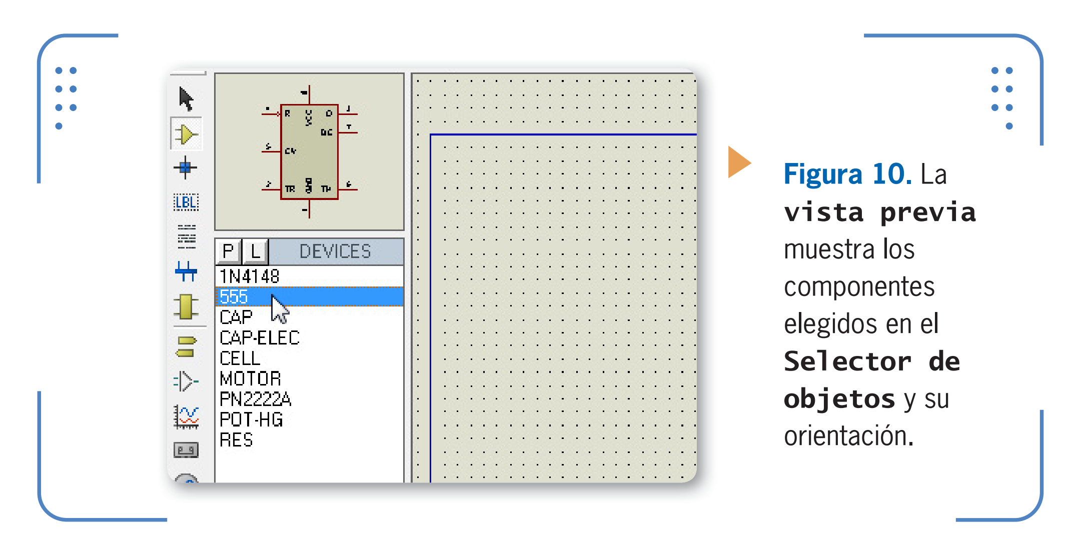


Figura 10. La **vista previa** muestra los componentes elegidos en el **Selector de objetos** y su orientación.

Una vez que tenemos todos los componentes necesarios en el **Selector de objetos** y sabemos cómo rotarlos y reflejarlos, ya podemos colocarlos en la ventana de edición para empezar a construir un circuito. A continuación, en el siguiente **Paso a paso**, explicaremos detalladamente cómo es el procedimiento.



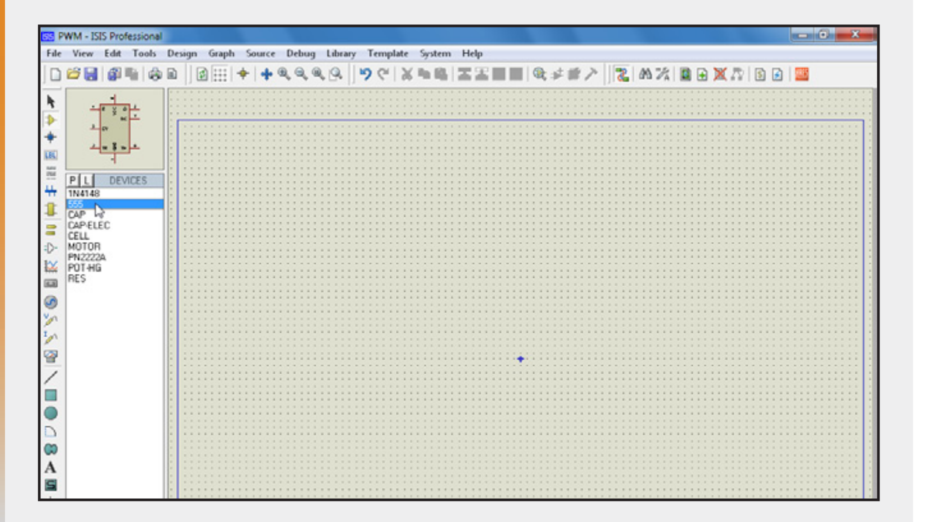
REEMPLAZAR COMPONENTES

Si bien trabajar con Proteus es algo sencillo, en algunas ocasiones debemos prestar mucha atención. Cuando intentamos colocar un componente en el mismo lugar donde se encuentra otro, Proteus nos preguntará si lo que queremos hacer es reemplazarlo. Al seleccionar la opción **OK**, el componente nuevo ocupará el lugar del anterior, pero debemos tener cuidado al hacerlo, ya que las conexiones cambiarán o desaparecerán si el componente nuevo es diferente del original.

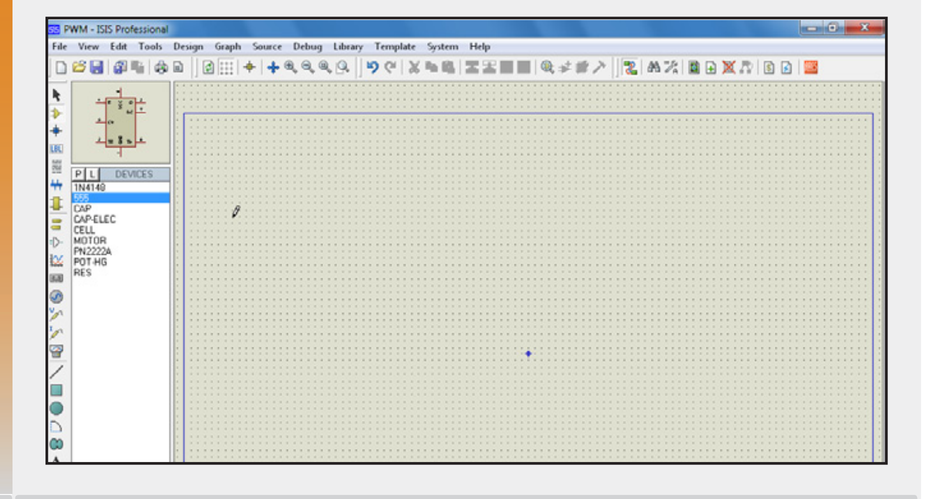
PASO A PASO: COLOCAR UN COMPONENTE



01 Seleccione el componente elegido en la ventana del Selector de objetos.

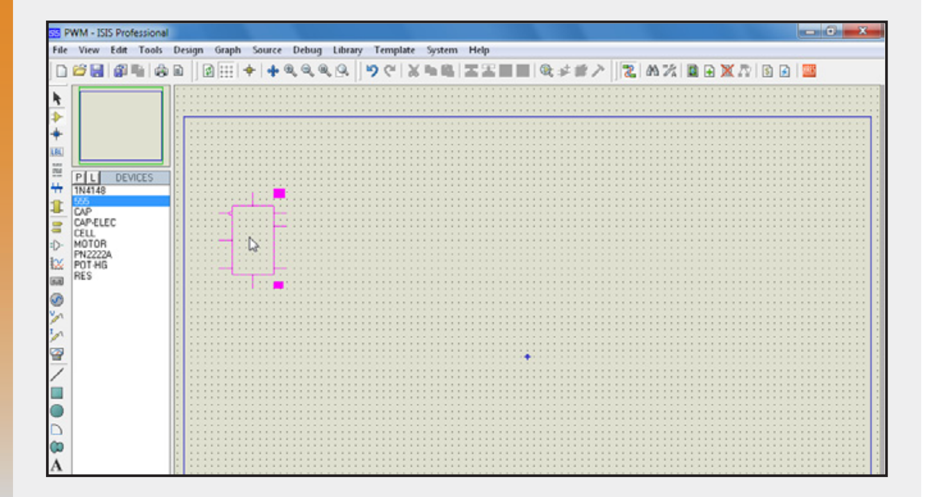


02 Coloque el cursor del mouse en algún lugar de la ventana de Edición; observe cómo este se convierte en un pequeño lápiz de color blanco.



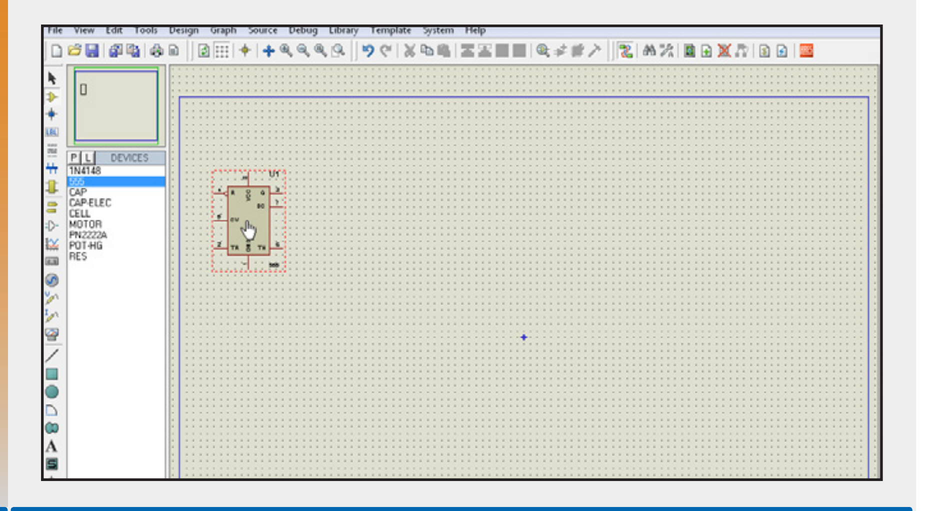


Haga clic en algún lugar vacío de la ventana de `Edición`; aparecerá la silueta del componente en color rosa. Puede mover el mouse, y la silueta del componente también se moverá hasta que elija un lugar donde colocarlo. Note que la silueta se alinea automáticamente con la rejilla mientras la mueve.



04

Una vez que haya decidido en qué lugar colocarlo, haga un clic, y el componente quedará ubicado en el área elegida.



Manipular los componentes en la ventana de Edición

Después de colocar varios componentes en la ventana de **Edición**, podemos cambiarlos de lugar, borrarlos, rotarlos o reflejarlos. Para seleccionar un componente que ya está en dicha ventana, simplemente hacemos un clic sobre él con el mouse, y este aparecerá con los bordes de color **rojo**, lo que significa que está seleccionado. Luego, para moverlo, hacemos un clic sobre él y, manteniendo el botón del mouse presionado, lo desplazamos hacia otro lugar.

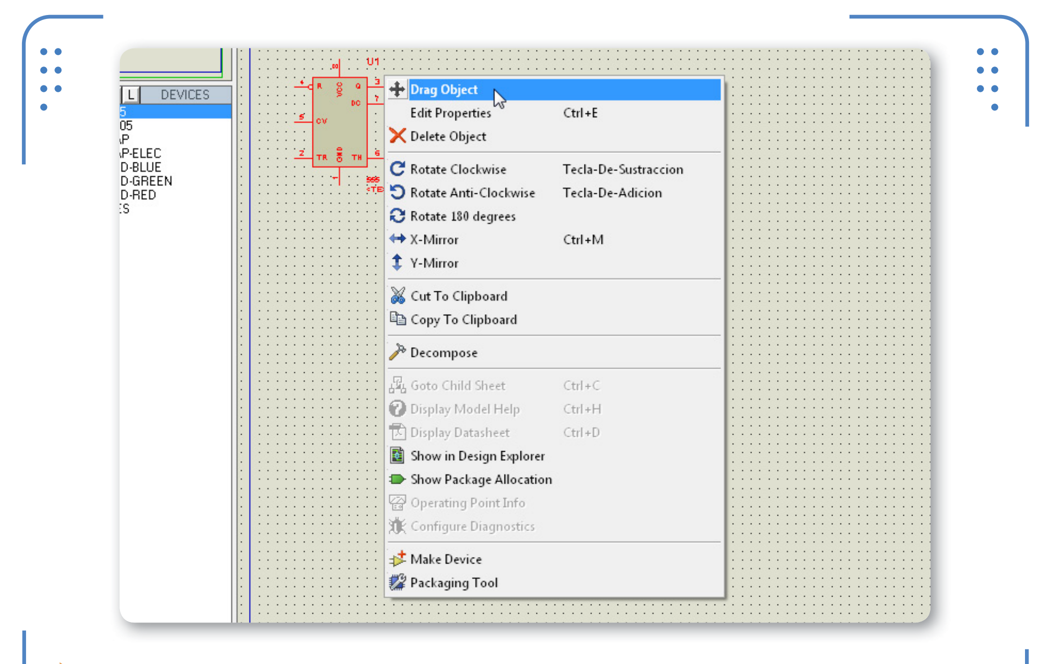


Figura 11. El menú contextual que se abre con un clic derecho tiene múltiples opciones para manejar los componentes.

Si hacemos un clic con el botón derecho del mouse sobre un componente, se abrirá un menú contextual con opciones que nos permiten mover, rotar, reflejar, borrar el componente, y algunas acciones adicionales que estudiaremos más adelante.

Mediante la opción **Drag Object**, movemos el componente a otro lugar, arrastrándolo y haciendo un clic en la ubicación elegida.

Las opciones de rotación y reflexión son idénticas a las de la barra de **Rotación y reflexión**, pero, como vimos, los botones de la barra no funcionan con los componentes ya colocados en la ventana de **Edición**. Entonces, si necesitamos rotar o reflejar un componente, debemos hacerlo desde este menú contextual. Aquí aparece una opción para rotar el componente 180 grados (**Rotate 180 degrees**), que no está en la barra de **Rotación y reflexión**.

La opción **Delete Object** borrará el componente de la ventana de **Edición**. También podemos eliminar un componente seleccionándolo y presionando la tecla **SUPR** en el teclado. Un atajo para borrar rápidamente un componente es hacer doble clic sobre él con el botón derecho del mouse.

Es importante tener en cuenta que, al seleccionar e intentar arrastrar un componente para moverlo, debemos asegurarnos de hacer clic en el cuerpo del componente y no en los textos que lo acompañan. Si hacemos clic sobre los textos, solo arrastraremos estos elementos.

Es posible seleccionar los componentes desde muchas de las opciones de la barra de herramientas de **Modo**, pero lo más adecuado es hacerlo desde el **Modo de selección**.

LO MÁS ADECUADO PARA SELECCIONAR COMPONENTES ES HACERLO DESDE EL MODO DE SELECCIÓN



Para seleccionar más de un componente a la vez, hacemos un clic en algún lugar vacío de la ventana de **Edición** y, manteniendo el botón del mouse presionado, lo arrastramos para crear un recuadro. Al soltar el botón del mouse, todos los elementos que se encuentren dentro de este recuadro quedarán seleccionados. Otra manera es mantener presionada la tecla **CTRL** mientras hacemos un clic sobre cada uno de los componentes que queremos seleccionar.



EL BOTÓN DERECHO DEL MOUSE



El botón derecho del mouse contiene algunos trucos que nos serán de gran utilidad en nuestra tarea. No solo sirve para acceder a los menús contextuales, sino que también podemos hacer doble clic con él sobre los componentes y las líneas de conexión para borrarlos; incluso es posible realizar una selección múltiple con este botón en vez de hacerlo con el izquierdo.

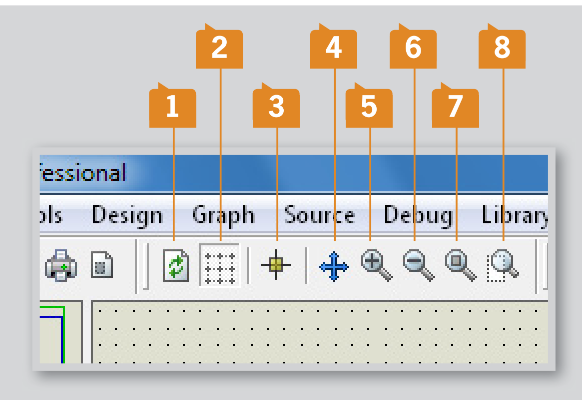
Si hacemos un clic con el botón derecho del mouse en un lugar vacío de la ventana de **Edición**, aparecerá un menú contextual que contiene la opción **Select All Objects**, para seleccionar todos los componentes.

Las herramientas de Visualización

A continuación, conoceremos en detalle las diferentes herramientas que integran la barra de **Visualización**.

GUÍA VISUAL ■ HERRAMIENTAS DE VISUALIZACIÓN





01 **REDRAW DISPLAY (REDIBUJAR PANTALLA):** redibuja la pantalla para actualizar la visualización del diseño. Debemos pulsar este botón si creemos que algo no se está mostrando de manera adecuada.

02 **TOGGLE GRID (CAMBIAR REJILLA):** con este botón podemos activar y desactivar la rejilla o cambiarla. Al presionarlo, modificaremos la rejilla de la siguiente manera: puntos-líneas-desactivada.

03 **TOGGLE FALSE ORIGIN (COLOCAR UN ORIGEN FALSO):** sirve para definir un origen falso para las coordenadas de la pantalla. Por defecto, el origen está en el centro de la hoja de trabajo indicado por un círculo y una cruz de color azul. Con este botón elegiremos otro punto como origen, si lo necesitamos. Al definir un origen falso, los valores en la barra de coordenadas cambiarán a color violeta para indicarlo.

04 **CENTER AT CURSOR (CENTRAR EN EL CURSOR):** este botón permite centrar el diseño en un nuevo punto. Al presionarlo, aparecerá un nuevo cursor en forma de un cuadro con una cruz; si hacemos clic en algún lugar de la ventana de **Edición**, esta se centrará en ese punto. ▶



ZOOM IN (ACERCAR): hace un acercamiento al diseño. Cada vez que lo presionemos, el circuito se hará más grande para poder ver alguna parte con más detalle.



ZOOM OUT (ALEJAR): realiza un alejamiento del diseño cada vez que lo pulsamos, para ver un área mayor del diseño.



ZOOM TO VIEW ENTIRE SHEET (VER LA HOJA COMPLETA): este botón es muy útil porque, al presionarlo, automáticamente se centrará y se mostrará la hoja de trabajo completa en la ventana de Edición.



ZOOM TO AREA (ACERCAR ÁREA): permite acercar un área determinada. Cuando presionamos este botón, el cursor tomará la forma de un cuadro con una cruz. Al hacer clic y arrastrar, dibujaremos un cuadro; con un nuevo clic, el área seleccionada se mostrará completa en la ventana de **Edición**. Es muy útil si queremos centrar y acercar una zona específica del diseño.

Las herramientas de **Vista** nos permiten acercar, alejar y centrar un área específica del diseño para verla o trabajar cómodamente. El acercamiento o alejamiento también puede hacerse con la rueda del mouse; al girarla hacia abajo, alejaremos la vista en la ventana de **Edición**, y al girarla hacia arriba, haremos un acercamiento. Estos movimientos se harán basados en la posición del cursor del mouse dentro de la ventana de **Edición**.

Las herramientas de Edición

La barra de herramientas de **Edición** también nos permite trabajar con los componentes en el diseño. En la siguiente **Guía visual** conoceremos los principales elementos que contiene.

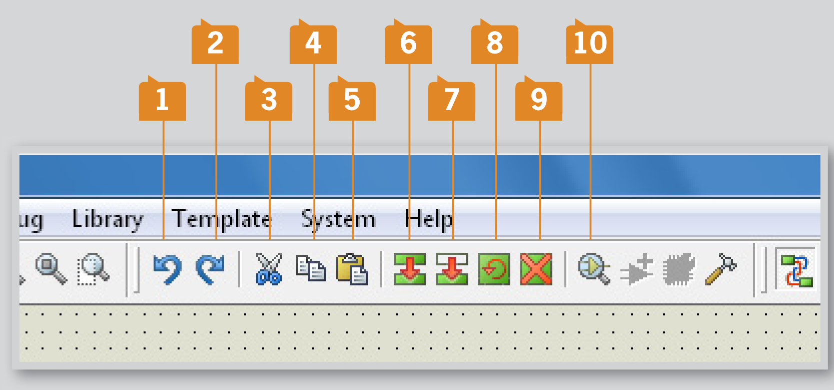


DIFERENCIA ENTRE COPIAR Y COPIAR BLOQUE

Cuando necesitamos copiar componentes o bloques de componentes en un diseño, siempre debemos hacerlo utilizando la opción **Block copy**. Si, en cambio, lo hacemos mediante la opción **Copiar** al Portapapeles de Windows, al pegarlos, las referencias de los componentes estarán repetidas, y esto generará un error cuando tratemos de simular el circuito.



GUÍA VISUAL ■ HERRAMIENTAS DE EDICIÓN



01 **UNDO CHANGES (DESHACER):** es el clásico botón de deshacer. Cuando hemos hecho un cambio y queremos revertirlo, podemos usar este botón.

02 **REDO CHANGES (REHACER):** rehace una acción que hemos revertido anteriormente con el botón deshacer.

03 **CUT TO CLIPBOARD (CORTAR):** podemos cortar y guardar en el Portapapeles algún elemento o varios de ellos que tengamos seleccionados.

04 **COPY TO CLIPBOARD (COPIAR):** permite copiar y guardar en el Portapapeles el o los elementos seleccionados.

05 **PASTE FROM CLIPBOARD (PEGAR):** pega en el diseño los elementos que tengamos en el Portapapeles de Windows.

06 **BLOCK COPY (COPIAR BLOQUE):** permite hacer una copia rápida de un bloque. Si seleccionamos uno o más elementos en la ventana de **Edición**, al presionar este botón, aparece una copia que podemos mover, y al hacer clic en algún lugar, los elementos quedan copiados allí. Es posible hacer varias copias consecutivamente, moviendo y haciendo clic en diferentes lugares, hasta que presionamos el botón derecho del mouse para terminar las copias.

07 **BLOCK MOVE (MOVER BLOQUE):** es similar al botón de copiar bloque, pero solo mueve los elementos seleccionados a otro lugar.

08 **BLOCK ROTATE (ROTAR BLOQUE):** permite rotar o reflejar los elementos seleccionados. Al presionarlo, aparece la ventana **Block Rotate/Reflect**, donde debemos ingresar un ángulo de rotación en el cuadro **Angle**, o podemos marcar alguna de las opciones **Mirror X** o **Mirror Y** para reflejar tanto en el eje X como en Y. Incluso, es posible usar las tres opciones al mismo tiempo. Nuevamente, los ángulos de rotación deben ser múltiplos de 90 grados.

09 **BLOCK DELETE (BORRAR BLOQUE):** borra del diseño todos los elementos seleccionados, aunque también podemos hacerlo presionando la tecla **SUPR** en el teclado.

10 **PICK PARTS FROM LIBRARIES (ELEGIR PARTES DE LAS LIBRERÍAS):** abre la ventana **Pick Devices** para elegir componentes de las librerías, de la misma forma en que lo hacemos con el botón **P** del **Selector de objetos**.

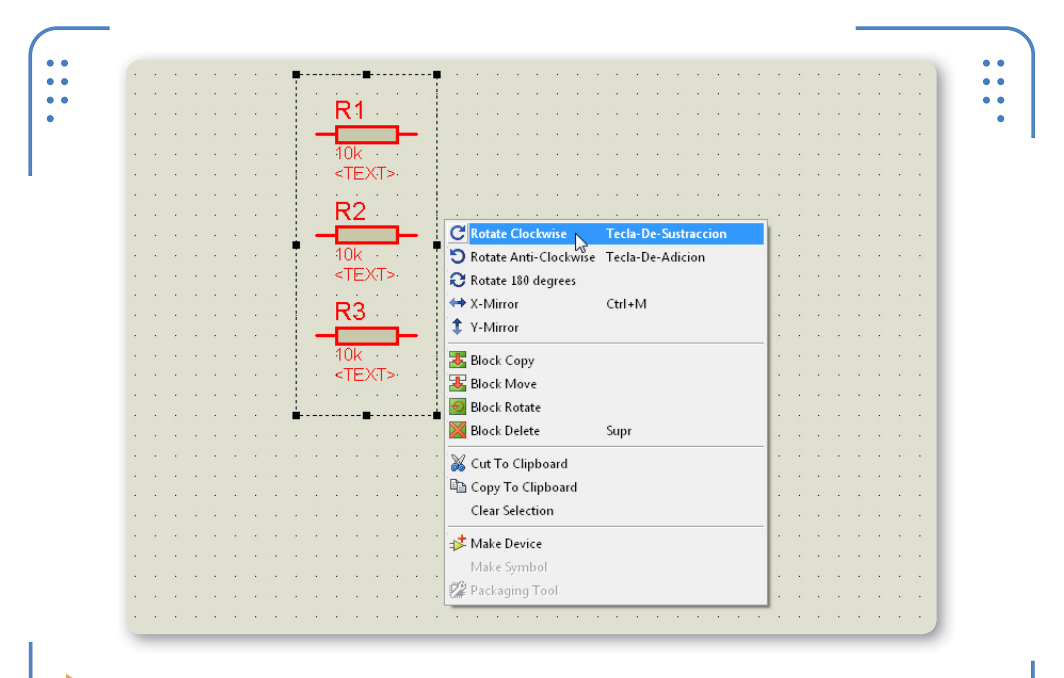
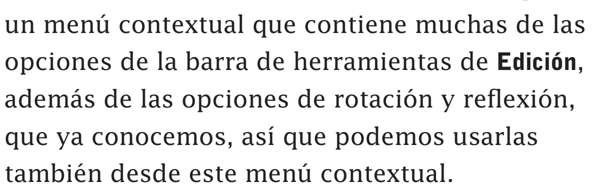


Figura 12. Al tener seleccionados varios componentes, el menú contextual muestra diferentes opciones de edición.

Como podemos apreciar, las diferentes herramientas de edición nos permiten trabajar con bloques de componentes. La selección de

componentes puede contener solo uno, varios de ellos o incluso todos los componentes del diseño. Si seleccionamos varios componentes y hacemos un clic con el botón derecho del mouse sobre ellos, aparece



un menú contextual que contiene muchas de las opciones de la barra de herramientas de **Edición**, además de las opciones de rotación y reflexión, que ya conocemos, así que podemos usarlas también desde este menú contextual.

> BLOCK COPY PERMITE COPIAR TODA UNA SECCIÓN DE NUESTRO DISEÑO DE MANERA SIMPLE

Entre ellas encontramos una que se destaca: el botón para copiar bloques o **Block Copy**. Este puede resultar de gran utilidad en aquellos diseños que contienen bloques que se repiten, ya que nos permite copiar toda una sección de nuestro diseño de manera simple y rápida, en vez de tener que dibujar todo el bloque otra vez. Como podemos ver, las herramientas de Proteus siempre simplifican nuestro trabajo.



Conexiones

Después de haber aprendido a colocar y manejar los componentes dentro de la ventana de **Edición**, es fundamental saber cómo se realizan las **conexiones** entre ellos dentro de un circuito. Para realizar las conexiones, debemos estar en el modo **Componente**, aunque es posible efectuarlas desde otros modos también.

En el siguiente **Paso a paso**, conoceremos en detalle cómo es el procedimiento para conectar componentes.



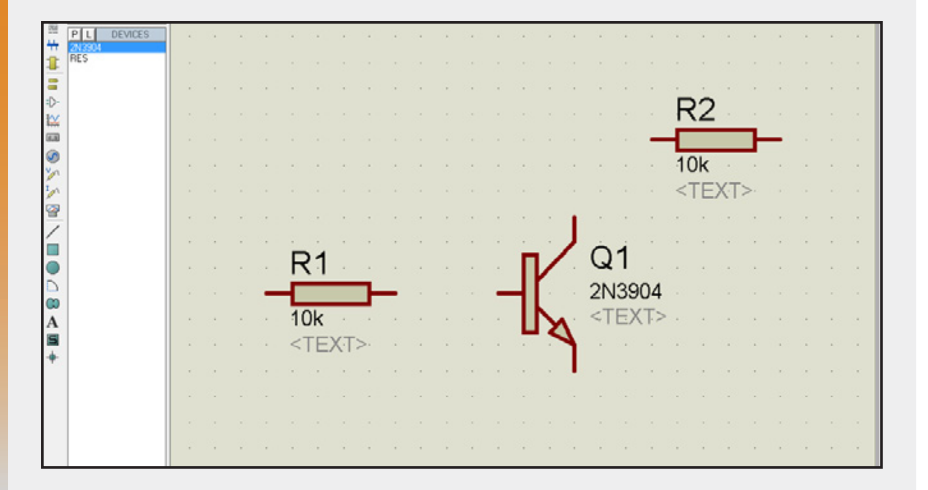
GRÁFICOS OPEN GL

Si recién instalamos Proteus en nuestra computadora, al abrir ISIS por primera vez puede aparecer una advertencia que nos informa que nuestra tarjeta gráfica soporta gráficos **Open GL** con aceleración por hardware. Esta clase de gráficos agrega funcionalidad y mejora la apariencia de la interfaz del módulo ISIS. Para activarlos debemos ingresar al menú **System** y seleccionar la opción **Set Display Options**. Es un procedimiento muy sencillo, que nos dará muy buenos resultados.

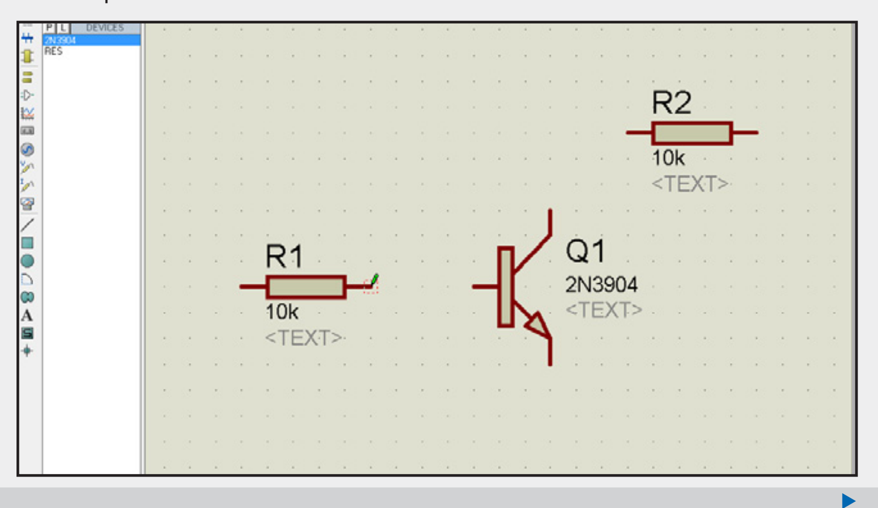
PASO A PASO: REALIZAR CONEXIONES



01 Para comenzar, debe tener ubicados algunos componentes en la ventana de Diseño, como, por ejemplo, un transistor y un par de resistores.

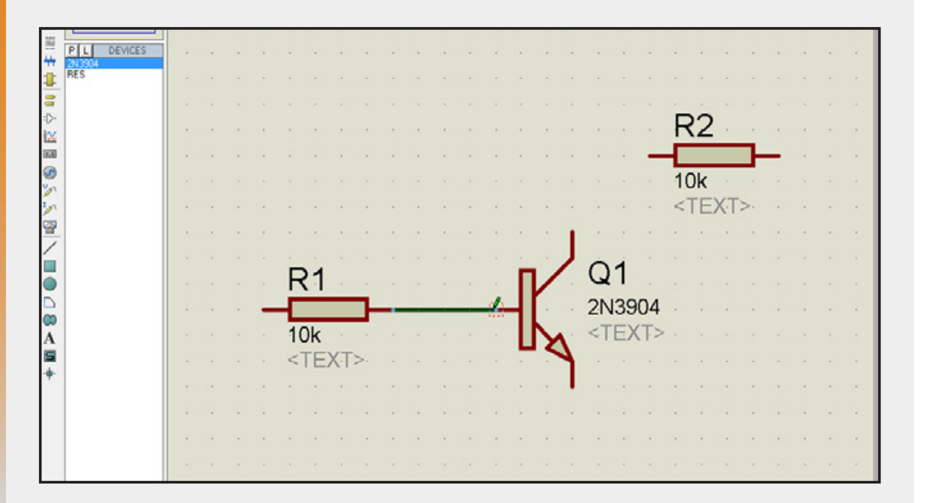


02 Luego, entre en el modo Componente. Coloque el cursor en la punta de la terminal del primer resistor; observe que el cursor toma la forma de un lápiz de color verde y aparece un cuadro de color rojo en la terminal. Esto indica que está en el lugar correcto para hacer una conexión.



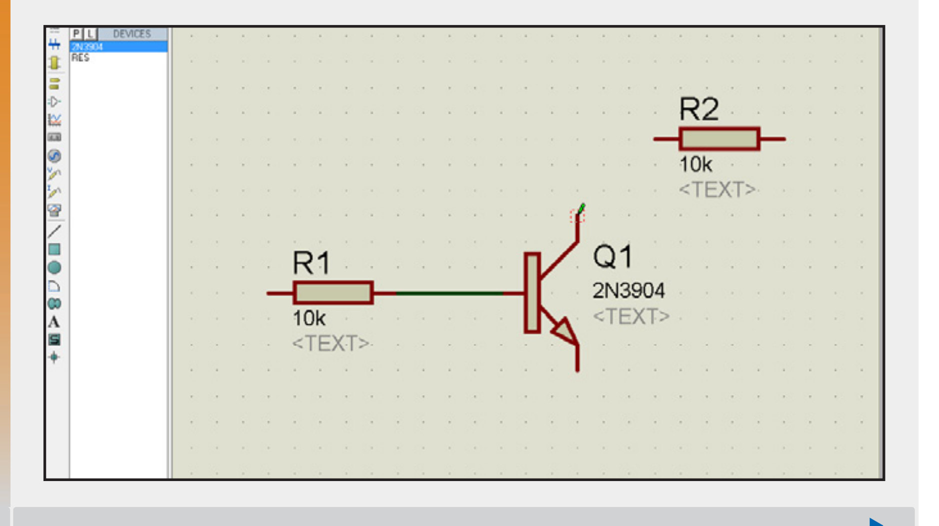
03

Haga un clic en ese lugar y desplace el cursor del mouse hacia la terminal donde se hará la conexión, en este caso, a la base del transistor. Al llegar a la terminal del transistor, el cursor volverá a tomar la forma de un lápiz verde, y un pequeño cuadro rojo indicará que se puede hacer la conexión. Haga un clic y la conexión quedará hecha.

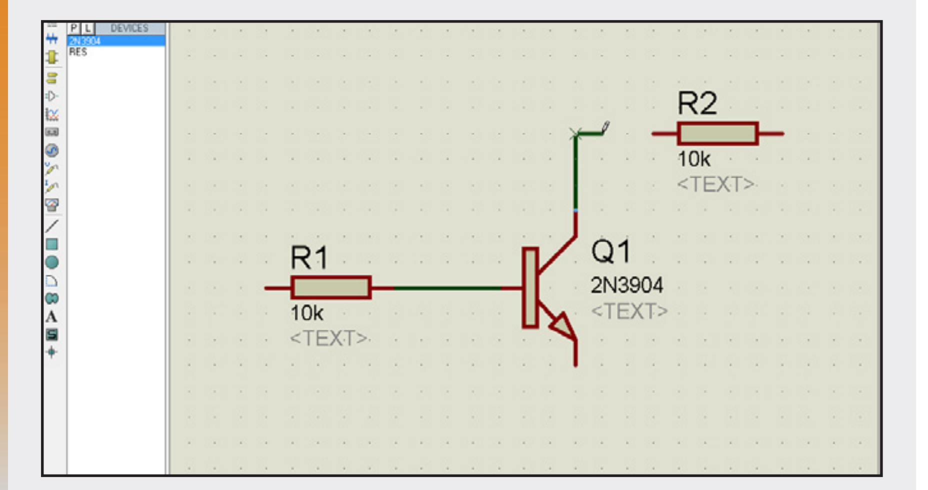


04

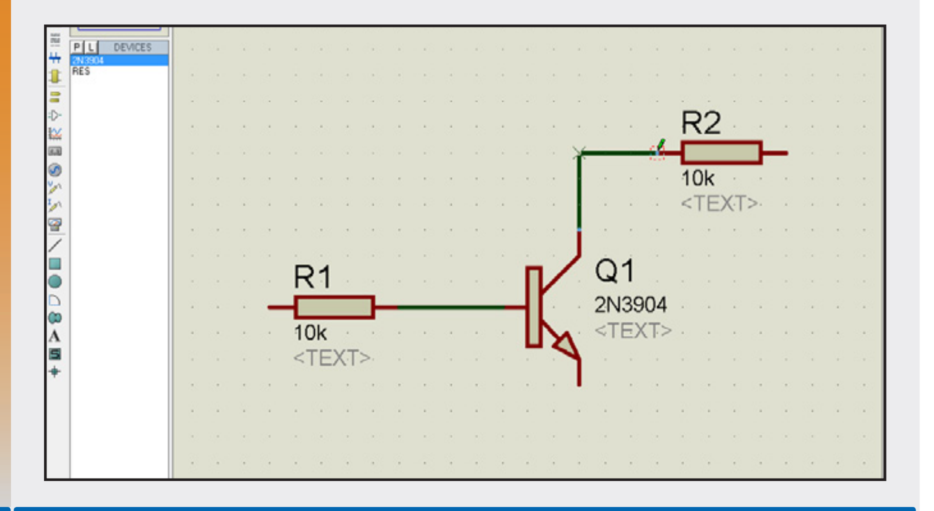
Para hacer la conexión entre el colector y el otro resistor, lleve el cursor a la terminal del colector del transistor (también puede empezar por la terminal del resistor, si lo desea).



05 Haga un clic para comenzar la conexión y desplace hacia arriba, hasta la altura del resistor. Puede hacer un clic en ese punto para indicar un cambio de dirección o trasladar la conexión directamente a la terminal del resistor, y Proteus calculará automáticamente la trayectoria de la conexión.



06 Al alcanzar la terminal del resistor, haga un clic, y la conexión quedará completa.



Como podemos notar, hacer conexiones en ISIS es simple y rápido. Podemos hacer así las conexiones entre las terminales de los componentes, y también es posible llevar las conexiones de una terminal a una línea de conexión ya existente o de una línea a otra.

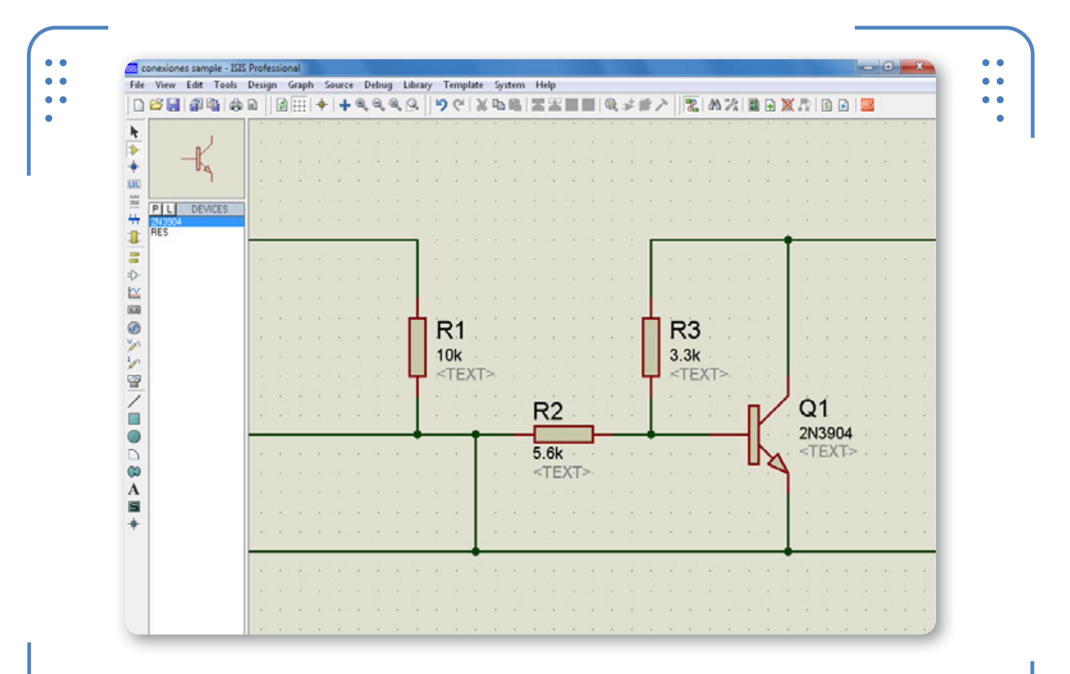


Figura 13. Conexiones entre componentes, entre un componente y una línea de conexión, o entre dos líneas de conexión.

UN PUNTO DE UNIÓN ES LO QUE INDICA QUE HAY CONEXIÓN ENTRE LÍNEAS QUE SE CRUZAN



Como vemos en la **Figura 13**, es posible conectar dos líneas entre sí. En el lugar donde se juntan las líneas, aparecerá un punto de unión para indicar que hay conexión entre ellas. En cambio, si las líneas se cruzan sin haber conexión entre ellas, el punto no estará presente, para mostrar que ahí no existe ninguna conexión.

Al realizar las diferentes conexiones dentro de un circuito, Proteus cuenta con una función de **autoruteo**, es decir que la trayectoria de la conexión es calculada de manera automática, mientas llevamos la conexión de un punto hacia otro.

Para activar o desactivar esta función usaremos el botón **Toggle Wire Autorouter**, que está en la barra de herramientas de **Diseño**, o podemos hacerlo desde el menú **Tools**, donde aparece la misma opción. Si desactivamos el autoruteo, solo se dibujará una línea recta entre los puntos donde hagamos la conexión, a menos que vayamos haciendo un clic en los puntos donde queremos fijar la conexión o realizar cambios de dirección. Es decir, nosotros decidiremos completamente la dirección que seguirá una conexión.

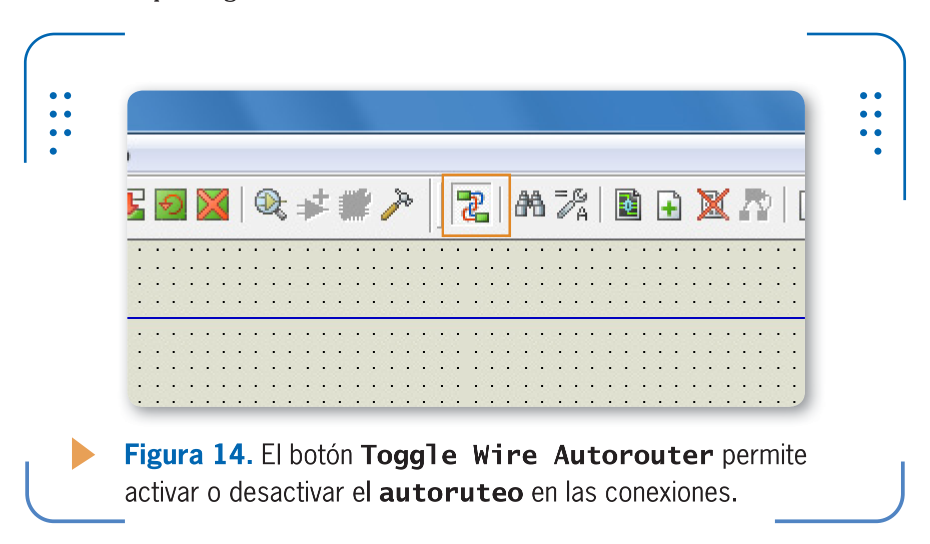


Figura 14. El botón **Toggle Wire Autorouter** permite activar o desactivar el **autoruteo** en las conexiones.

Mover y borrar conexiones

Después de realizar las conexiones, podemos moverlas para ajustarlas y así mantener el diseño ordenado. Para esto, debemos ingresar en el modo de **Selección**. Veamos cómo hacerlo.



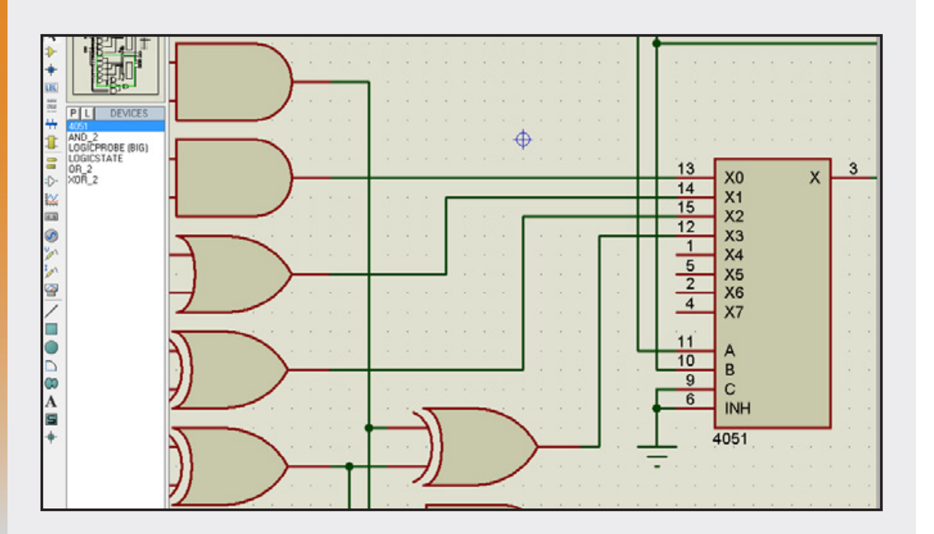
DIFERENTES FORMAS DE CONECTAR

Como podemos imaginar, en Proteus existen diferentes maneras para conectar los componentes. No siempre realizaremos las conexiones entre componentes llevando líneas de conexión entre sus terminales o pines. Veremos que también es posible usar otros elementos, comos **buses** y **terminales** para hacer conexiones sin líneas e, incluso, entre diferentes hojas de trabajo. En los próximos capítulos, estudiaremos estos temas en profundidad.

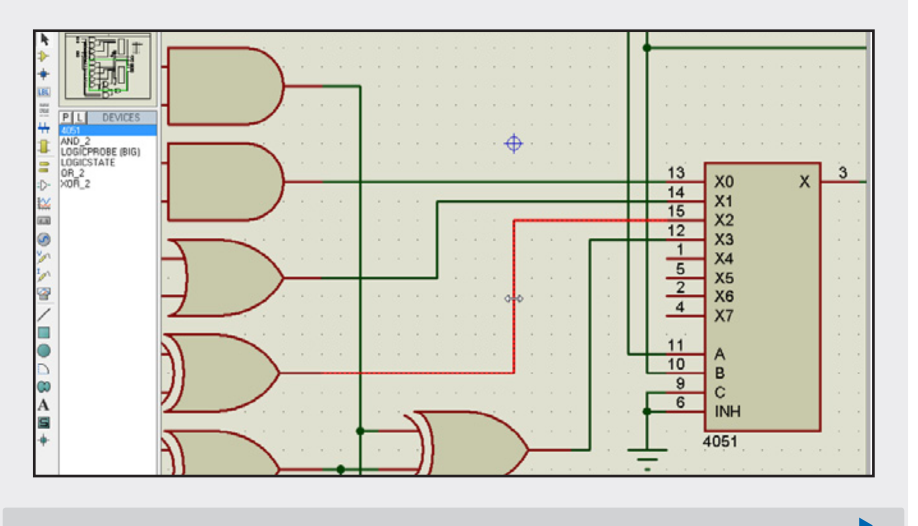
PASO A PASO: MODIFICAR CONEXIONES



01 Debe tener un diseño con conexiones realizadas y estar en el modo de **Selección**.



02 Seleccione la línea de conexión por modificar; esta quedará en rojo. Si coloca el cursor sobre ella, este tomará la forma de una línea horizontal o vertical con dos flechas según la orientación del segmento; o de una cruz si está sobre una esquina.



03 Haga un clic sobre la línea seleccionada y arrastre el cursor en la dirección deseada sin soltarlo: a la izquierda o derecha para un segmento vertical, hacia arriba o abajo para uno horizontal, o en cualquier dirección si es una esquina de la conexión. Suelte el botón del mouse para que la línea o la esquina tomen su nueva posición.

Mediante este sencillo procedimiento, reorganizaremos las conexiones para que todo quede ordenado como queremos. También es posible seleccionar más de una línea de conexión a la vez y moverlas juntas, para acelerar la tarea.

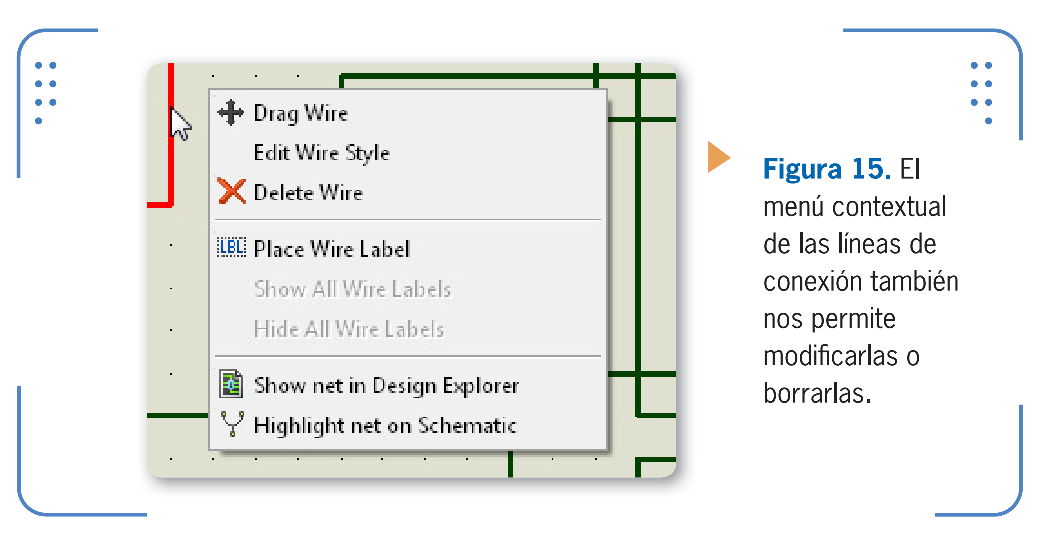


Figura 15. El menú contextual de las líneas de conexión también nos permite modificarlas o borrarlas.

Además, podemos hacer un clic con el botón derecho del mouse sobre una de las líneas de conexión para abrir un menú contextual, en el cual tendremos algunas opciones para editar dichas líneas.

Drag Wire permite arrastrar y mover una línea, y **Delete Wire** permite borrarla. Otra manera de eliminar una línea de conexión rápidamente es hacer un doble clic con el botón derecho del mouse sobre ella.



Dibujar un circuito

En el siguiente **Paso a paso**, veremos un ejemplo sumamente sencillo de cómo se dibuja un circuito en ISIS. Para hacerlo, vamos a partir de un circuito muy simple: un **oscilador** formado por una compuerta NOT con disparador Schmitt.

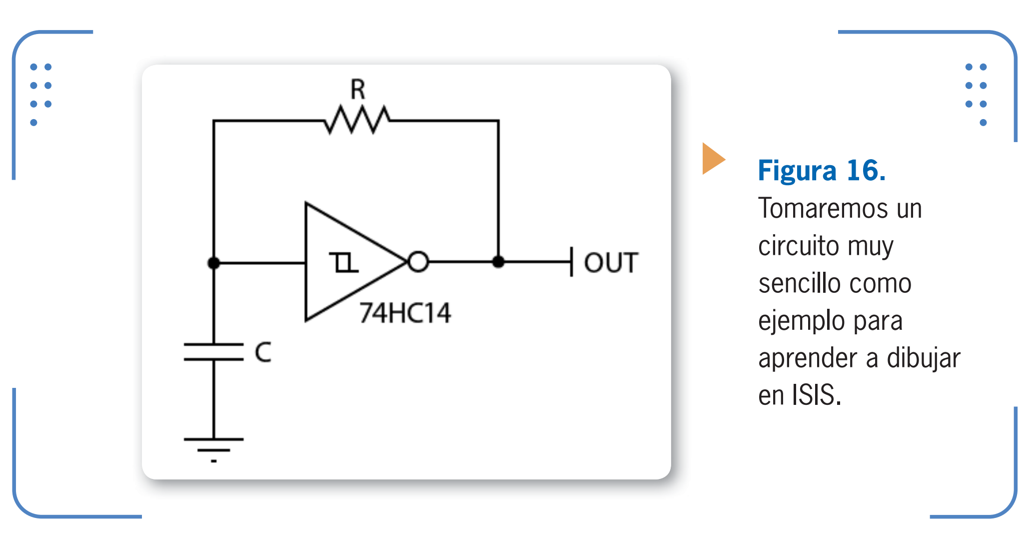


Figura 16. Tomaremos un circuito muy sencillo como ejemplo para aprender a dibujar en ISIS.



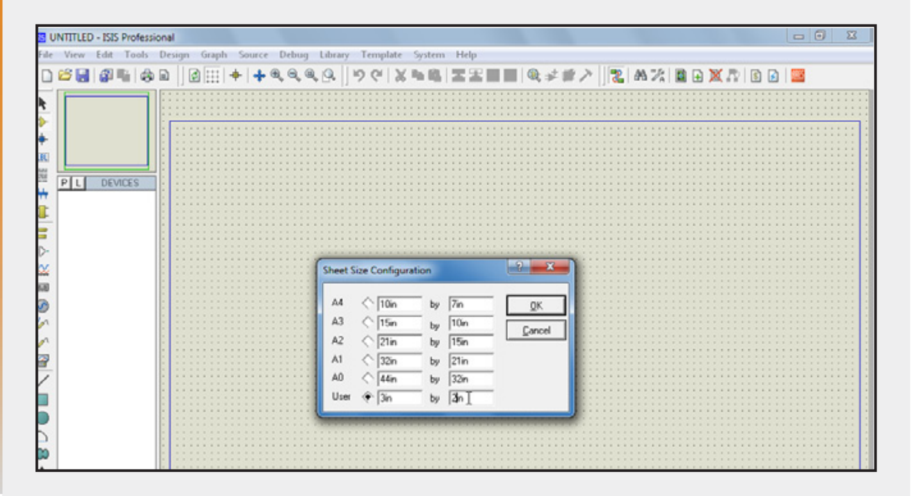
OPEN GL CONTRA GDI

ISIS ofrece dos modos gráficos. Por un lado, **GDI**, en donde la apariencia y los efectos visuales están controlados por Windows. Por otro lado, **Open GL**, en donde los efectos están controlados por la tarjeta gráfica de la computadora. Este último modo agrega algunas mejoras, por ejemplo, animación al seleccionar los componentes en la ventana de ISIS.

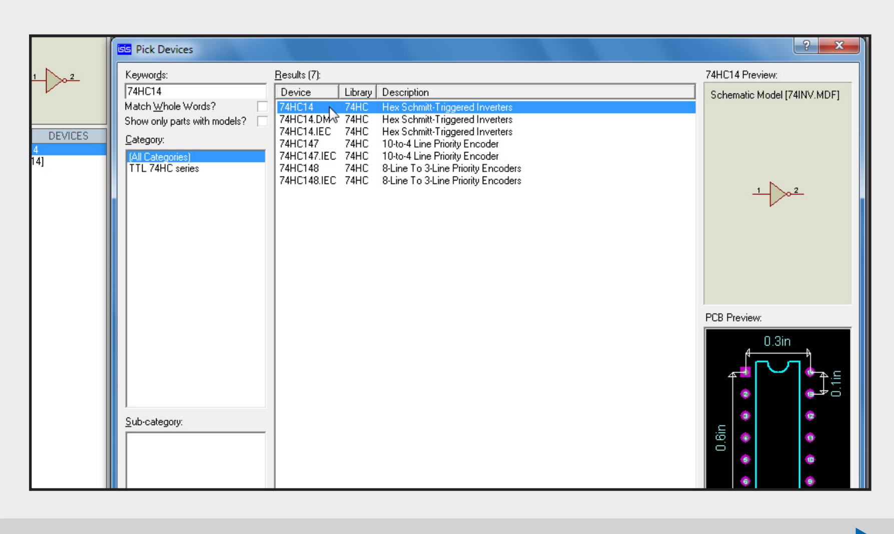
▼ PASO A PASO: DIBUJAR UN CIRCUITO EN ISIS



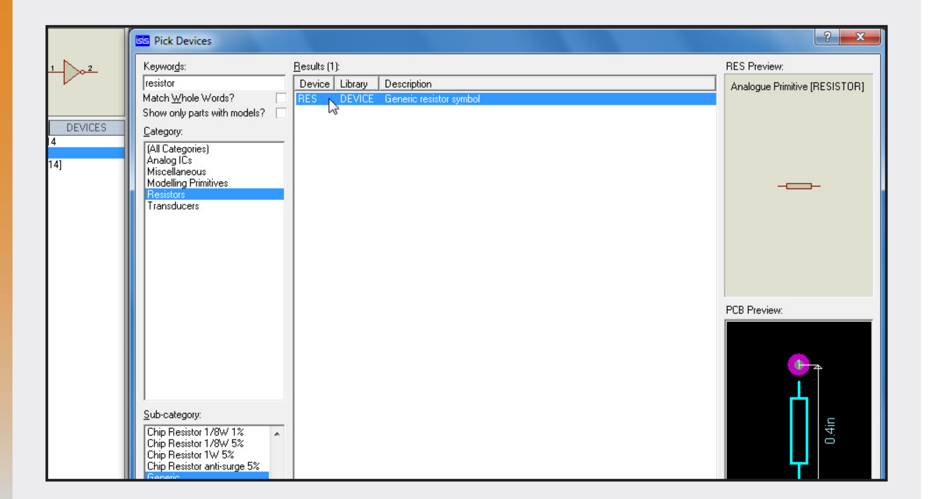
01 Abra ISIS desde su icono para tener un nuevo documento listo para dibujar el circuito, y cambie el tamaño de la hoja de trabajo desde el menú `System/Set Sheet Sizes`... a **3in x 2in**. Haga esto aquí para mejorar la visualización.



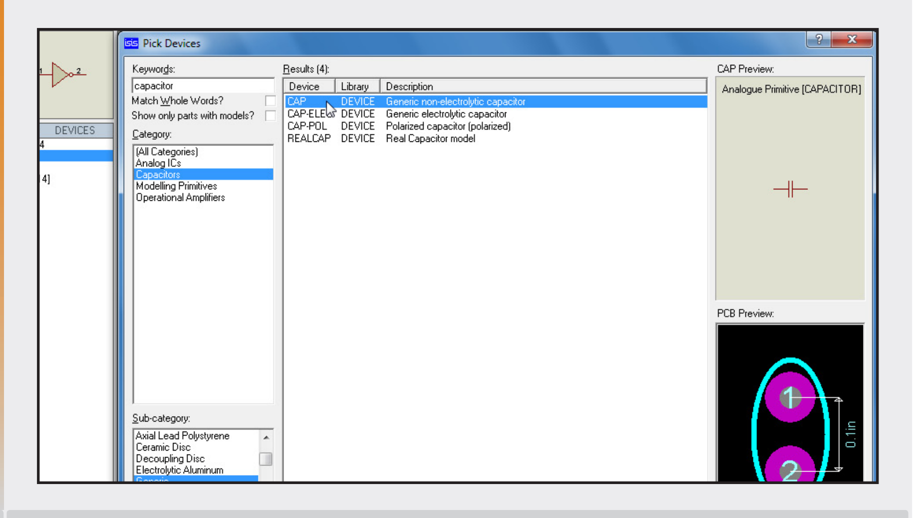
02 Vaya al modo `Componente` y presione el botón `P` para abrir la ventana `Pick Devices`, desde donde va a elegir los componentes necesarios. En este caso, busque y seleccione un `74HC14`.



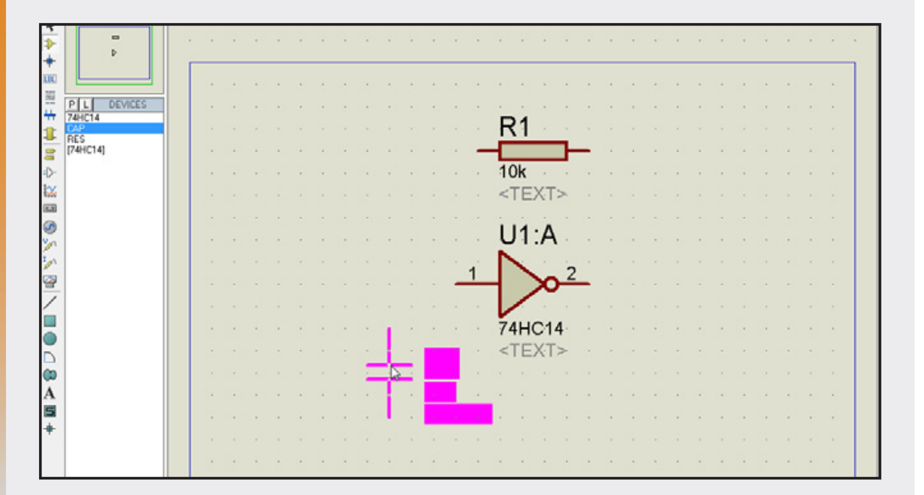
03 Ingrese en el cuadro de búsqueda la palabra **resistor**; en la ventana Category, seleccione Resistors y en Sub-category, busque y seleccione Generic. Esto le permitirá seleccionar un resistor genérico.



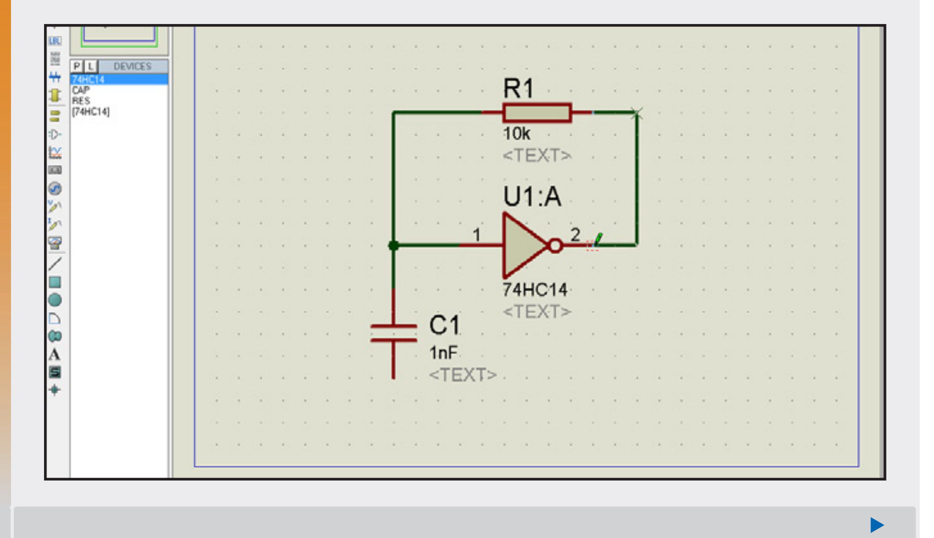
04 Ahora escriba en el cuadro de búsqueda la palabra **capacitor**; en la ventana Category, seleccione Capacitors y en Sub-category, busque y seleccione otra vez Generic. Esto le permitirá seleccionar un capacitor genérico. Una vez elegidos los componentes, puede cerrar la ventana Pick Devices.



05 Coloque los componentes en la ventana de `Edición`, como se ve en el circuito de referencia. Tal vez tenga que rotar alguno de ellos para ubicarlo en la posición correcta. Como ya vimos, puede hacerlo antes de colocarlo, con las herramientas de `Rotación` o en el menú contextual, con el componente ya posicionado en la ventana de `Edición`.

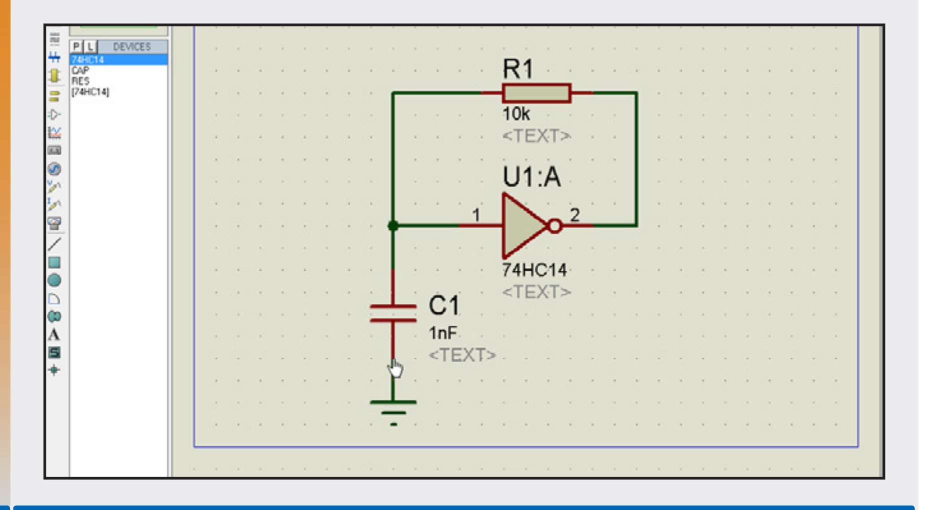


06 Desde el modo `Componente`, haga las conexiones correspondientes entre ellos.



07

Presione el botón para entrar en el modo de `Terminales`; en la lista, seleccione la terminal GROUND, colóquela en el diseño debajo del capacitor y conéctela con él.



Hemos dibujado nuestro primer circuito en ISIS; ahora veamos cómo guardarlo en la computadora.

Guardar un diseño

Para guardar un archivo con el diseño que acabamos de dibujar, debemos ir al menú **File/Save Design As...** y, en la ventana **Save ISIS Design File**, elegir una carpeta y colocar un nombre al circuito.

Los diseños en ISIS se almacenan con la extensión **.DSN**. Si hacemos doble clic en cualquier archivo de este tipo, se abrirá automáticamente ISIS con el diseño que corresponde.

Otros tipos de archivos

Además del archivo **.DSN**, ISIS genera otros de forma automática y en la misma carpeta, al estar trabajando y guardar el diseño mediante el comando **Save Design**. Estos poseen la extensión **.DBK** y son archivos de respaldo. Por ejemplo, para un diseño de ISIS llamado **Circuito.dsn**,

se genera un archivo **Last loaded Circuito.dbk** o **Backup Of Circuito.dbk**. El **.DSN** contiene siempre la última versión guardada del diseño, en tanto que el **Backup Of** tiene una versión previa a la última guardada, y **Last Loaded**, la última versión del diseño abierta exitosamente. Cada vez que guardamos el diseño, los archivos de respaldo se actualizan de manera automática, para tener siempre un respaldo disponible en caso de necesitarlo. Como estos archivos tienen la misma extensión, siempre encontraremos solo uno de ellos a la vez.

Podemos usar los archivos **.DBK** para restaurar un diseño que se ha perdido o dañado. Para abrir estos archivos, vamos al menú **File/Open Design...** y, en la parte inferior, en la lista desplegable **Tipo**, elegimos la opción **Backup Design Files**, para que se muestren en la ventana de exploración.

Durante la edición de un circuito también pueden crearse archivos con el mismo nombre del diseño y con la extensión **.PWI**; estos guardan el estado de algunas ventanas auxiliares que se abren en ISIS y algunas configuraciones.

ADEMÁS DEL ARCHIVO .DSN, ISIS GENERA OTROS DE FORMA AUTOMÁTICA Y EN LA MISMA CARPETA



La ventana de Vista previa

La pequeña ventana que está arriba del **Selector de objetos** puede ayudarnos a navegar por los circuitos, especialmente si estos son grandes. Cuando estamos trabajando en la ventana de **Edición**, en la ventana de **Vista previa** tenemos una imagen en miniatura del diseño, que se corresponde con la hoja de trabajo completa.

Además, visualizamos un **cuadro de color verde** que indica el área que aparece en la ventana de **Edición**. Esto nos sirve de referencia para ver en qué sección del diseño o de la hoja de trabajo nos encontramos.



CUÁNDO USAR LA OPCIÓN TIDY

La opción **Tidy** puede resultar muy útil para mantener los diseños limpios y ordenados, pero es recomendable usarla solo al final, cuando el diseño esté completo, ya que si limpiamos el **Selector de objetos** antes, quizá borremos componentes que sí planeamos usar, pero aún no hemos empleado.

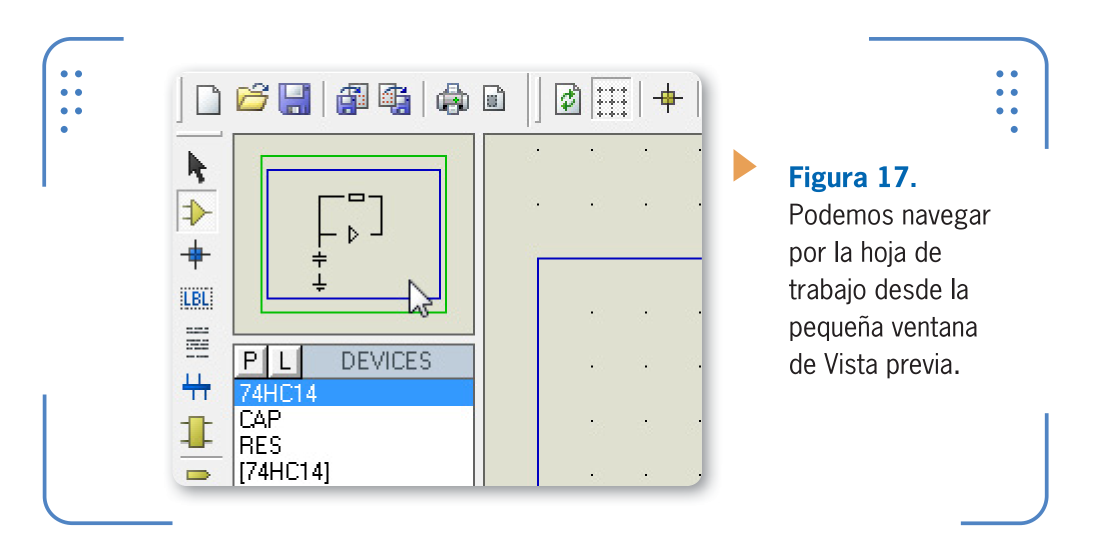


Figura 17. Podemos navegar por la hoja de trabajo desde la pequeña ventana de Vista previa.

Podemos hacer un clic en cualquier lugar de esta ventana de **Vista previa**, que nos llevará a esa área centrada en la ventana de **Edición**. Además, el cursor toma la forma de una cruz con cuatro flechas, lo que determina que podemos mover el cuadro verde para desplazarnos a otra zona. Así, navegaremos por la hoja de trabajo desde la ventana de **Vista previa**, hasta que, al hacer un nuevo clic, la zona elegida quedará centrada en la ventana de **Edición**.

Eliminar componentes no utilizados

ISIS cuenta con una función para limpiar los diseños y retirar los componentes no utilizados. Se denomina **Tidy**, y podemos encontrarla en el menú **Edit**. Al seleccionarla, se nos pedirá confirmación para llevar a cabo la acción. Cuando presionamos el botón **OK** suceden dos cosas: primero, se eliminan todos los componentes en la ventana de **Edición** que se encuentran fuera de la hoja de trabajo; y segundo, se eliminan los componentes de la lista en el **Selector de objetos** que no hayan sido utilizados en el diseño.

LA FUNCIÓN TIDY RETIRA LOS COMPONENTES NO UTILIZADOS EN LOS DISEÑOS



Esta operación es muy útil para limpiar nuestro diseño y el **Selector de objetos**. Si hacemos clic con el botón derecho del mouse sobre el Selector, mientras estamos en el modo **Componente**, también

obtendremos la opción **Tidy**. En este caso, al elegirla, solo se borrarán aquellos objetos no usados en el **Selector de objetos**, pero los componentes que estén fuera de la hoja de trabajo en la ventana de **Edición** no se eliminarán.

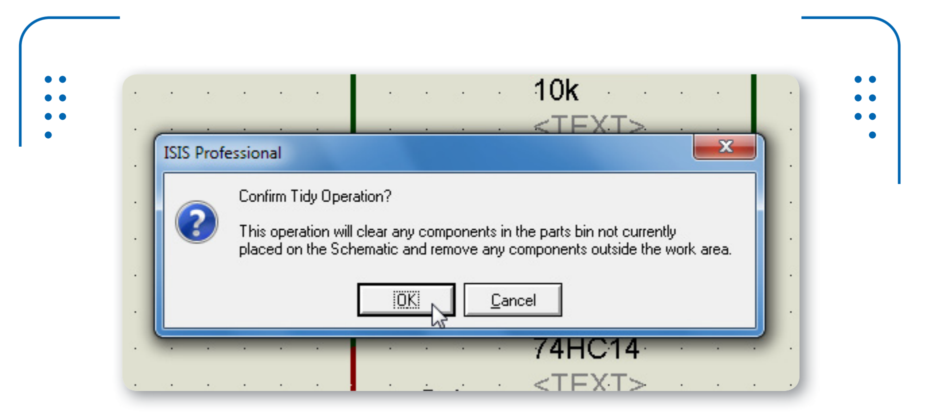


Figura 18. **Tidy** permite mantener nuestro diseño limpio y ordenado al eliminar los componentes que no se usan.



RESUMEN

En este capítulo inicial, hemos aprendido a utilizar las herramientas principales de ISIS. Nos familiarizamos con su interfaz y conocimos las bases para dibujar circuitos electrónicos en Proteus. Esto es muy importante porque, para simular cualquier circuito, primero debemos dibujarlo. También es fundamental saber crear circuitos si solo queremos usar Proteus como herramienta para el dibujo de diagramas. En el siguiente capítulo, veremos algunas herramientas de dibujo adicionales, y comenzaremos a estudiar las funciones de simulación con ejemplos y proyectos.

Actividades

TEST DE AUTOEVALUACIÓN

1	¿Qué significa VSM?
2	¿Cuál es la versión más reciente de Proteus disponible en el mercado?
3	¿En cuántos idiomas se distribuye Proteus?
4	¿Cómo se cambia el tamaño de la hoja de trabajo?
5	¿Dónde está ubicada por defecto la barra de herramientas de **Modo**?
6	¿Cómo se accede a la ventana **Pick Devices**?
7	¿Cómo se rota un componente colocado en la ventana de **Edición**?
8	¿Cómo se borra un componente ubicado en la ventana de **Edición**?
9	¿Cómo podemos copiar un grupo de componentes?
10	¿Qué extensión tienen los archivos de ISIS?

ACTIVIDADES PRÁCTICAS

1	Abra la ventana **Pick Devices** y navegue por las diferentes categorías para familiarizarse con ellas y con los componentes que Proteus tiene disponibles.
2	Busque un circuito sencillo y dibújelo en ISIS para practicar.
3	Busque y dibuje un nuevo circuito en ISIS. Esta vez vaya al menú **File/New Design...** y elija una plantilla diferente de la **DEFAULT**.
4	Guarde en su computadora los circuitos dibujados.
5	Vaya al menú **Help/Sample Designs**, y navegue por los archivos de ejemplo que contiene Proteus.



Introducción a la simulación en Proteus

En este segundo capítulo, continuaremos estudiando algunas opciones de dibujo de diagramas electrónicos en el módulo ISIS. Además, comenzaremos a aprender cómo se simula un circuito electrónico en Proteus, y entenderemos las bases de la simulación analógica, digital y mixta. Presentaremos algunas herramientas de análisis, que serán sumamente útiles, junto con las opciones de animación.

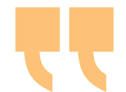




Por qué simular circuitos electrónicos

La simulación de circuitos electrónicos puede ser útil en diferentes niveles: desde aplicaciones didácticas para que los novatos aprendan el funcionamiento de componentes y circuitos básicos, pasando por los aficionados que desean diseñar o construir sus propios circuitos, hasta llegar al nivel profesional, para el desarrollo de proyectos y productos electrónicos comerciales.

PODEMOS VERIFICAR EL FUNCIONAMIENTO DE UN CIRCUITO DE MODO VIRTUAL



Proteus es todo un **laboratorio virtual** de electrónica, donde podemos dibujar un circuito y verificar si realmente funciona como esperamos. Nos permite contar con una gran cantidad de componentes, herramientas e instrumentos virtuales destinados a generar todo tipo de señales y usarlas en los circuitos, además de disponer de un completo grupo de instrumentos de medición para analizar las señales que manejarán o generarán los circuitos.

De esta forma, es posible verificar completamente el funcionamiento y las señales en un circuito de modo virtual, sin tener que soldar ni conectar un solo cable. Podemos usar múltiples herramientas de análisis, que en la realidad son costosas y quizá no estén a nuestro alcance, todas las veces que necesitemos en el simulador.

El motor de simulación analógica de Proteus es **ProSPICE**, que está basado en el software **SPICE**, desarrollado por la Universidad de Berkeley, con el estándar **SPICE3f5**.

El motor para la simulación digital es **DSIM**, que en combinación con ProSPICE, permite simular circuitos mixtos, es decir, una parte digital interactuando con una parte analógica.

Los dos motores de simulación, **ProSPICE** y **DSIM**, están totalmente integrados al módulo ISIS, lo cual hace correr las simulaciones de una manera muy sencilla. Es posible usar diversas opciones de animación, análisis e interactividad con muchos de los elementos del circuito, desde la misma ventana de ISIS, simplemente dibujando un circuito en la ventana de **Edición**.



Los componentes

Proteus cuenta con más de **34.000** componentes y partes, entre dispositivos genéricos, animados, componentes específicos, circuitos integrados, conectores, interruptores, herramientas de simulación, baterías y fuentes, microcontroladores, etcétera. Podemos elegir entre una gran lista de componentes para dibujar los circuitos; en la **Tabla 1** veremos un resumen de las principales categorías de componentes.

PRINCIPALES COMPONENTES

CATEGORÍA	DESCRIPCIÓN
Analog ICs	Circuitos integrados analógicos (filtros, reguladores de voltaje, amplificadores, etc.).
Capacitors	Todo tipo de capacitores (cerámicos, electrolíticos, etc.).
CMOS 4000 series	Circuitos integrados de la serie 4000 CMOS.
Connectors	Todo tipo de conectores (de audio, USB, headers, etc.).
Data converters	Convertidores (A/D, D/A y sensores de temperatura, entre otros).
Diodes	Todo tipo de diodos (rectificadores, zener, etc.).
Electromechanical	Diferentes tipos de motores.
Inductors	Bobinas y transformadores.
Memory ICs	Memorias (RAM, EPROM, EEPROM, etc.).
Microprocessor ICs	Microprocesadores, microcontroladores y periféricos.
Modelling primitives	Componentes genéricos primarios analógicos y digitales (compuertas resistores, transistores, etc.).
Operational amplifiers	Todo tipo de amplificadores operacionales.
Optoelectronics	Optoelectrónica (LEDs, displays, etc.).
Resistors	Todo tipo de resistores, genéricos y específicos.

Switches and relays	Interruptores, relevadores y teclados.
Switching devices	Dispositivos de conmutación (DIACs, TRIACs, etc.).
Transducers	Transductores (sensores de presión, distancia, etc.).
Transistors	Todo tipo de transistores, genéricos y específicos.
TTL 74xxx	Familia de circuitos integrados de las series 74.

Tabla 1. Principales categorías de componentes disponibles en Proteus.

Componentes simulables y no simulables

Dentro de los componentes de Proteus, encontramos una gran cantidad de elementos que tienen un **modelo de simulación**, es decir, que se pueden simular, y otros que carecen de este modelo y solo pueden ser usados para dibujar diagramas.

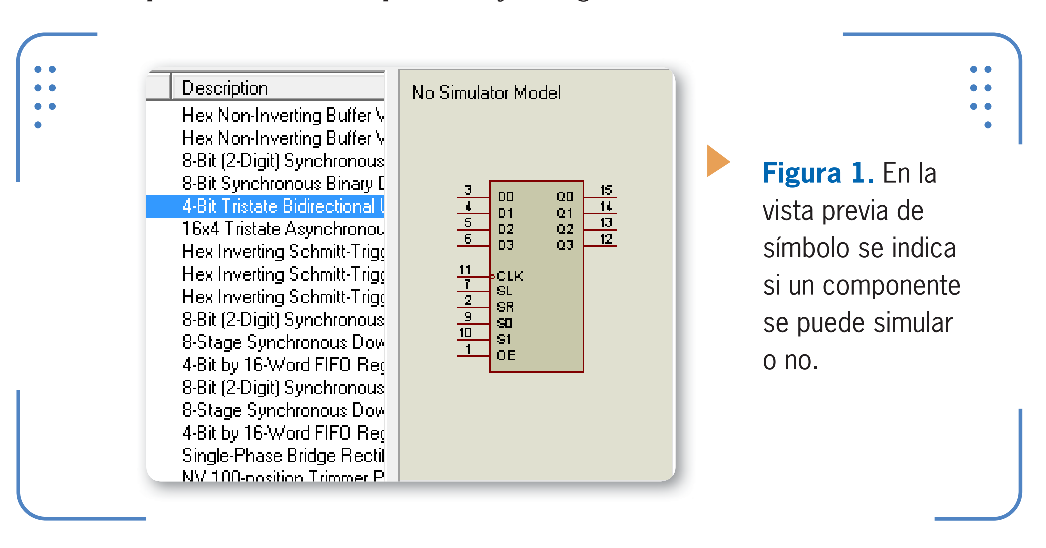


Figura 1. En la vista previa de símbolo se indica si un componente se puede simular o no.

Al navegar por las librerías de ISIS, dentro de la ventana **Pick Devices**, notaremos que, cuando seleccionamos un componente, aparecerá un texto en la parte superior de la ventana de Vista previa de símbolo, que

en algunos casos dice: **VSM DLL Model**, **Schematic model** o **SPICE model**, y en otros: **Analogue Primitive** o **Digital Primitive**. Esto indica que el componente se puede simular. En otros casos se mostrará la leyenda: **No Simulator Model**, para indicer que ese componente no tiene modelo y no puede formar parte de una simulación.

La librería Active y otros componentes especiales

En las librerías de ISIS, hallaremos algunos componentes especiales que serán de utilidad al simular circuitos, como en la librería **ACTIVE**.

Figura 2. Algunos de los principales elementos que integran la librería ACTIVE del módulo ISIS.

En la **Tabla 2**, que veremos a continuación, presentamos una lista de los principales componentes o herramientas de esta librería y una descripción de cada uno de ellos. Algunos son genéricos, mientras que otros son herramientas que facilitan las simulaciones.

LIBRERÍA ACTIVE

CATEGORÍA	DESCRIPCIÓN
BATTERY	Batería o pila, podemos especificar su voltaje.
ALTERNATOR	Alternador, genera una señal senoidal, debemos indicar su frecuencia y amplitud.
FUSIBLE	Fusible, es necesario que ingresemos la corriente nominal.
SPEAKER	Altavoz, debemos especificar el voltaje y la impedancia.
BUZZER	Zumbador, tenemos que ingresar voltaje de operación, frecuencia y resistencia.
SOUNDER	Altavoz piezoeléctrico interactivo.
LAMP	Lámpara incandescente, debemos indicar el voltaje de operación y la resistencia.
LED	LEDs de varios colores y de doble color.
TORCH_LDR	LDR y lámpara para simular diferentes condiciones de iluminación.
BUTTON	Pulsador interactivo, raramente se puede presionar durante las simulaciones.
KEYPAD	Teclados de diferentes tipos.
TOUCHPAD	Panel sensible al toque interactivo.
POT_HG	Potenciómetro, podemos cambiar su resistencia y comportamiento.
CAPACITOR	Capacitor animado, muestra su carga interna durante las simulaciones.
MOTOR	Motor genérico, debemos especificar el voltaje, la resistencia y las revoluciones.
FLIP FLOP D	Flip flop genérico tipo D.
FLIP FLOP JK	Flip flop genérico tipo JK.
CLOCK	Generador de reloj, es posible especificar la frecuencia de la señal.

Tabla 2. Algunos elementos de la librería ACTIVE.

Es importante destacar que los componentes **SPEAKER**, **BUZZER** y **SOUNDER**, al ser elementos que generan sonido, pueden escucharse a través de los altavoces de la computadora durante las simulaciones. Además de los componentes listados, existen varias clases de interruptores, compuertas lógicas genéricas, un par de relevadores genéricos y jumpers activos, es decir que se pueden simular.

NO TODOS LOS COMPONENTES QUE INTEGRAN LA LIBRERÍA DEVICE PUEDEN SIMULARSE



Más componentes

Otras librerías contienen más componentes o elementos de gran utilidad; por ejemplo, en **DEVICE** tenemos: un cristal (**CRYSTAL**), capacitores genéricos, otra batería (**CELL**), transistores genéricos y transformadores, entre otros. Debemos tener en cuenta que no todos los componentes de esta librería pueden simularse.

En a la ventana **Pick Devices**, dentro de la categoría **Modelling primitives** (modelos primarios), encontraremos otros modelos de simulación genéricos, pertenecientes a dos librerías: **ASIMMDLS** y **DSIMMDLS**, que corresponden a modelos analógicos y digitales, respectivamente.



Los controles de simulación

Una vez que tenemos un circuito dibujado en ISIS, podemos simularlo. Para hacerlo, utilizaremos algunos botones que se encuentran en la barra de herramientas de **Simulación** y que vamos a explicar en la siguiente **Guía visual**.

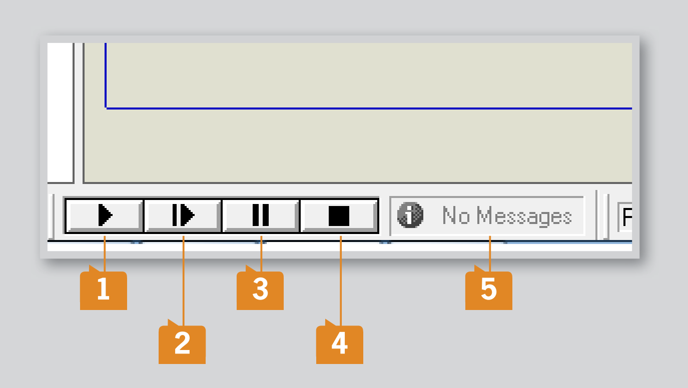


MODELOS PRIMARIOS

Existen diferentes tipos de componentes simulables, aunque la diferencia es interna y no debemos poner mayor atención en ella. Los modelos primarios (**Primitive**) son los que están ya incluidos dentro del motor de simulación **ProSPICE** y son los más básicos, como: resistores, capacitores, diodos y compuertas.

GUÍA VISUAL ■ LOS CONTROLES DE SIMULACIÓN





01 **PLAY (REPRODUCIR):** con este botón iniciamos la simulación del circuito que tenemos dibujado en ISIS.

02 **STEP (PASO):** permite ejecutar la simulación por pasos. Cada vez que lo presionamos, la simulación avanza un tiempo determinado y se pone en pausa automáticamente.

03 **PAUSE (PAUSA):** con este botón podemos pausar una simulación que se está llevando a cabo.

04 **STOP (DETENER):** detiene la simulación, sin importar si está corriendo o en pausa.

05 **LOG MESSAGES (MENSAJES):** en este cuadro encontraremos mensajes importantes acerca de la simulación. Podemos hacer un clic en él para que se abra un informe detallado acerca de la simulación en curso o la última realizada.

La primera simulación

Con lo que hemos aprendido hasta el momento, podemos comenzar a simular un circuito. El simple hecho de dibujar un circuito en ISIS nos permite simularlo, con tan solo hacer un clic en el botón **Play** (reproducir) de la barra de **Simulación**.

Como primer ejemplo, utilizaremos el circuito del oscilador que dibujamos en el **Capítulo 1**. Entonces, al presionar el botón **Play**, comenzará la simulación de manera continua, esto quiere decir que el oscilador empezará a funcionar.

Figura 3. Podemos apreciar algunos detalles especiales en la interfaz de ISIS al momento de simular un circuito.

En la ventana de ISIS, observaremos algunos cambios. El botón **Play** se muestra en color verde, indicando que la simulación está corriendo. En la barra de título, aparece el nombre del archivo y, entre paréntesis, la palabra **Animating**. En la barra de estado también encontramos **ANIMATING** y un reloj que cuenta el tiempo que ha pasado de la simulación (por defecto, esta correrá en tiempo real); además, se informa el porcentaje de uso de la CPU (**CPU Load**). En el cuadro de mensajes de la barra de **Simulación** veremos un símbolo y un número de mensajes; hablaremos de esto más adelante.

En el circuito aparecerán unos pequeños cuadrados de colores **azul**, **rojo** y **gris**; se trata de los niveles lógicos que tenemos en las diferentes partes del circuito. Notaremos que los cuadrados cambian de color constantemente, para indicar que el oscilador está funcionando. Por ahora, esto es lo que podemos observar en la simulación, pero poco a poco conoceremos más herramientas de visualización y análisis que nos ampliarán en gran medida la comprensión de las simulaciones. Para detener la simulación presionamos el botón **Stop**.

El informe de simulación

En el **cuadro de mensajes** que se encuentra ubicado en la barra de herramientas de **Simulación** podemos ver un reporte con los detalles del proceso. Cuando abrimos un circuito o creamos uno nuevo, en el cuadro encontraremos la leyenda **No Messages** (no hay mensajes). Al correr una simulación, la leyenda cambiará e indicará un número definido de mensajes. La cantidad es un valor que depende de cada simulación en particular y de lo que suceda en ella.

Si hacemos un clic en el cuadro cuando indica algún número de mensajes, se abre la ventana **SIMULATION LOG**, en donde podemos leer los mensajes en detalle. Es posible acceder a este informe tanto si la simulación está corriendo como si no lo está. Cuando está detenida, el reporte nos dará información del último proceso llevado a cabo.

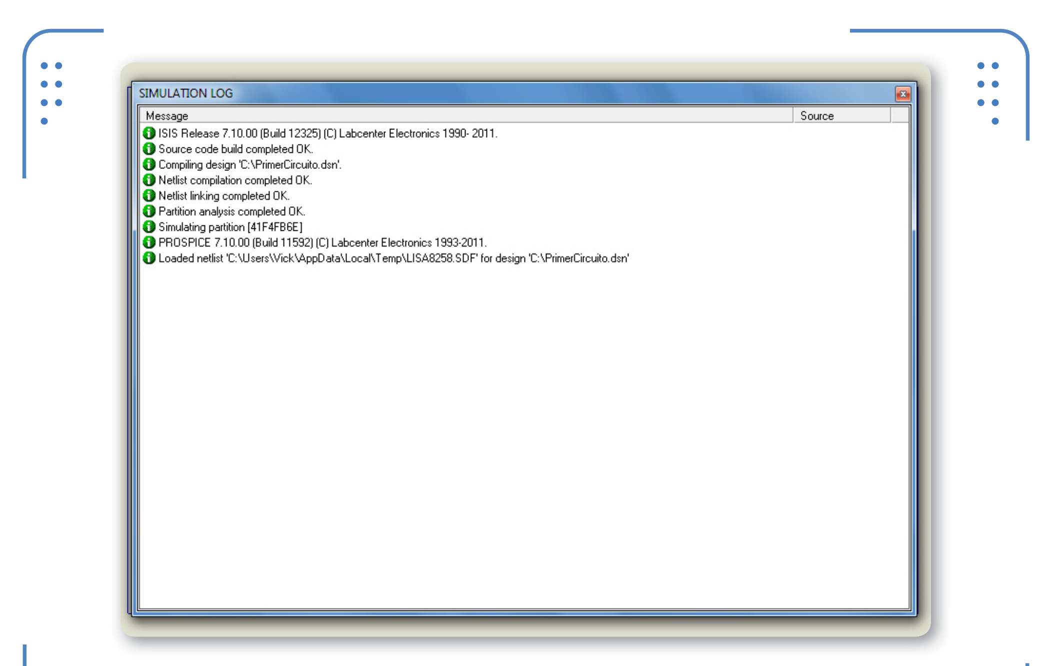


Figura 4. En este informe podemos observar los detalles de lo que sucede cuando corremos una simulación.

Si la simulación se desarrolla de manera exitosa, en el cuadro de mensajes veremos un icono de color verde con una letra **i** (**information**). En cambio, si hay alguna advertencia sobre algo que no

funciona bien, aparecerá un triángulo de color amarillo con un signo de exclamación (**!**), para indicarnos que hay mensajes a los que debemos prestarles atención. Por último, si al intentar correr una simulación ocurre algún error que impide que esta se inicie, o si al estar corriendo no puede continuar, la simulación se detendrá y se abrirá automáticamente la ventana con el informe, en el cual se indicarán los errores mediante un icono con una **x** y un texto explicativo, en color rojo. Con esta información sabremos cuáles son los errores y podremos tratar de corregirlos para que la simulación funcione correctamente.

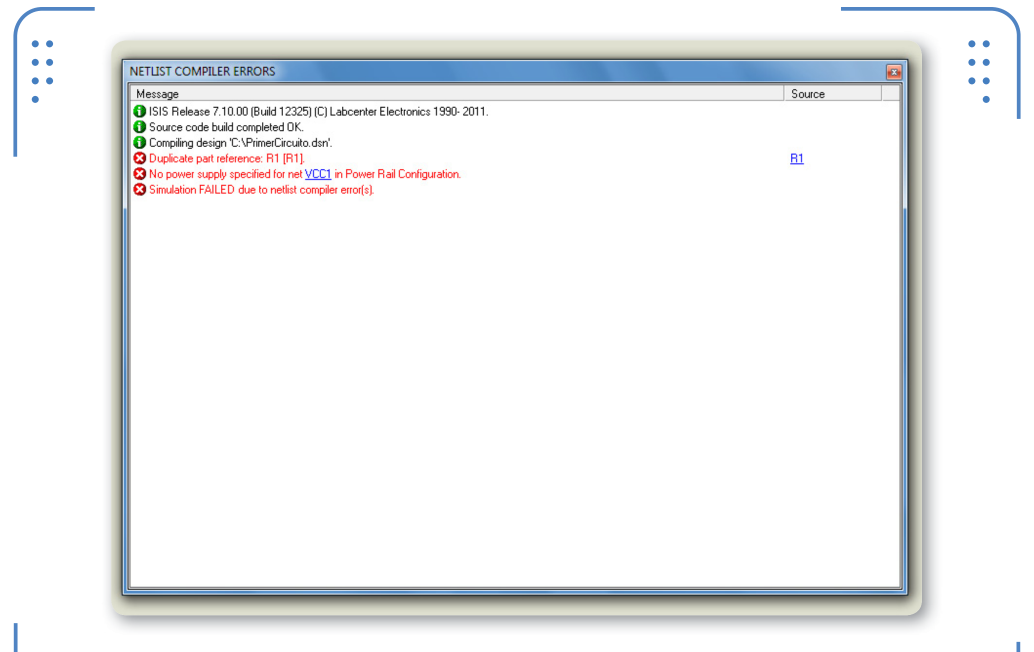


Figura 5. Los textos en rojo y con una **x** indican una falla por la cual la simulación no puede llevarse a cabo.



LOS BOTONES DURANTE LA SIMULACIÓN

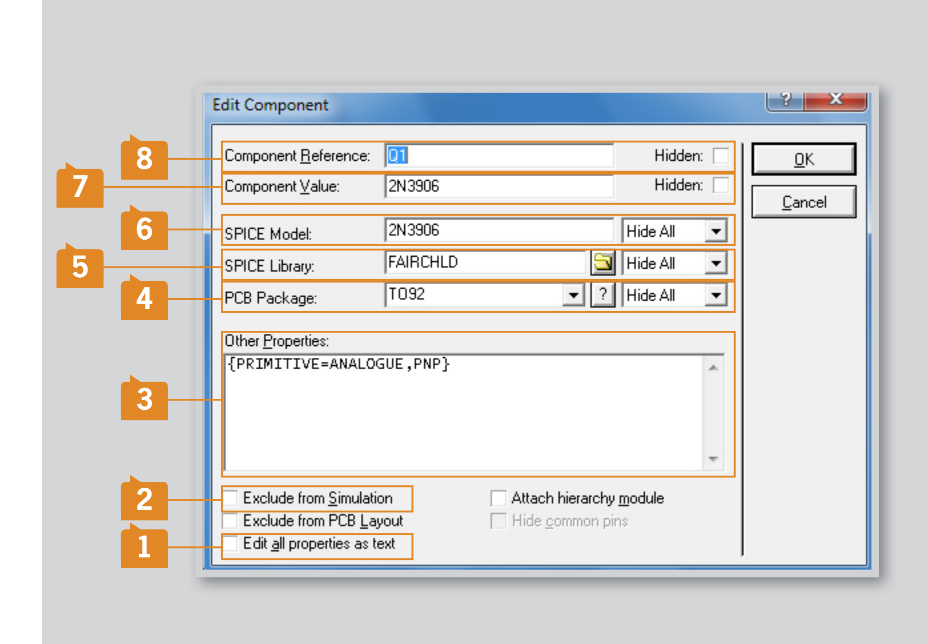
Algunos de los botones de las diferentes barras de herramientas cambiarán y se mostrarán en color gris durante la simulación. Esto se debe a que esas funciones o herramientas no pueden utilizarse mientras una simulación está corriendo. Si necesitamos usarlos, debemos detener la simulación.

Propiedades de los componentes

Cada componente tiene sus propiedades específicas, a las cuales podemos acceder de dos maneras. Una es hacer clic derecho del mouse sobre un componente colocado en la ventana de **Edición** y, desde el menú contextual, elegir la opción **Edit Properties**. La otra forma es dar un doble clic sobre el componente (o solo un clic si ya está seleccionado), para que se abra la ventana **Edit Component**. En la siguiente **Guía visual** la conoceremos en detalle.

GUÍA VISUAL ■ LA VENTANA EDIT COMPONENT





01 **EDIT ALL PROPERTIES AS TEXT (EDITAR TODAS LAS PROPIEDADES COMO TEXTO):** si marcamos esta opción, tendremos la posibilidad de editar las propiedades como texto. Al hacerlo, algunas de las opciones aparecerán como texto en la ventana **Other Properties** y podremos modificarlas desde ahí.



02 **EXCLUDE FROM SIMULATION (EXCLUIR DE LA SIMULACIÓN):** si seleccionamos esta casilla, el componente no se incluirá en la simulación, y será como si no estuviera presente en el circuito.

03 **OTHER PROPERTIES (OTRAS PROPIEDADES):** presenta propiedades adicionales que puede tener el componente. Como se muestran en formato texto, podemos editarlas, borrarlas o agregar otras, si lo necesitamos.

04 **PCB PACKAGE (EMPAQUE PARA CIRCUITO IMPRESO):** aquí veremos el empaque del componente; esto es especialmente útil si vamos a diseñar un circuito impreso para nuestro diagrama. En algunos casos, tendremos más de un empaque que podremos elegir de la lista desplegable.

05 **SPICE LIBRARY (LIBRERÍA SPICE):** en esta sección se indica la librería a la que pertenece el componente. Al igual que el modelo, no debemos cambiar este valor.

06 **SPICE MODEL (MODELO SPICE):** en este campo aparece el modelo de simulación. Por lo general, no debemos cambiar este dato, ya que identifica el modelo matemático que usará Proteus para simular el componente. También podemos mostrar u ocultar todo este campo o parte de él.

07 **COMPONENT VALUE (VALOR DEL COMPONENTE):** indica el valor, el nombre o el número de parte del componente; en este caso es **2N3906**, pero también podemos cambiarlo si lo deseamos. Al igual que la referencia, es posible ocultar este valor al marcar la casilla **Hidden**.

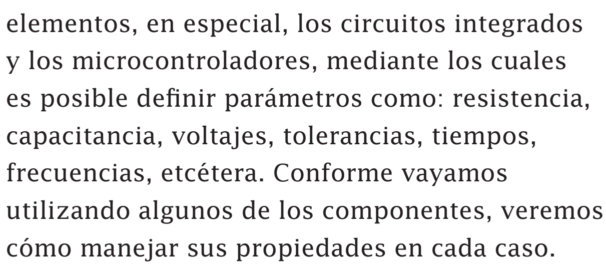
08 **COMPONENT REFERENCE (IDENTIFICADOR DEL COMPONENTE):** muestra la identificación del componente en el circuito. En este caso, como es un transistor, la referencia es **Q**, pero esto depende del componente de que se trate, y podemos cambiarla si lo deseamos. Si marcamos la casilla **Hidden** (oculto) de la derecha, esta propiedad no aparecerá en el diagrama junto al componente.



SCHEMATIC MODELS

Los **Schematic Models** son componentes complejos formados por elementos primarios para emular el funcionamiento de un componente específico, como un circuito integrado. Se forman principalmente por fuentes de corriente ideales, fuentes de voltaje e interruptores primarios, para hacerlos más eficientes.

Las propiedades de los diferentes componentes presentan un formato similar al mostrado anteriormente, aunque los campos dependen de cada caso en particular. Algunos podrán tener más elementos, en especial, los circuitos integrados y los microcontroladores, mediante los cuales es posible definir parámetros como: resistencia, capacitancia, voltajes, tolerancias, tiempos, frecuencias, etcétera. Conforme vayamos utilizando algunos de los componentes, veremos cómo manejar sus propiedades en cada caso.



ALGUNOS COMPONENTES PUEDEN TENER MÁS CAMPOS PARA CONFIGURAR

La mayoría de estas propiedades serán útiles al momento de realizar la simulación del circuito, ya que definen los valores y los parámetros de funcionamiento del componente. Algunas otras propiedades se utilizan solo para etiquetar los componentes dentro del circuito; por ejemplo, la referencia que se mostrará en el diagrama para identificar al componente dentro de él.

Editar las etiquetas de texto de los componentes

Es posible editar las etiquetas de los textos que acompañan a cualquiera de los componentes, por ejemplo, el **identificador** o el **valor**. Para lograrlo, debemos hacer doble clic sobre cualquier etiqueta y así abrir el cuadro de diálogo de edición.

En la pestaña **Label**, editamos el texto dentro del campo **String**, lo ubicamos en forma horizontal o vertical con la opción **Rotate**, y cambiamos la justificación en la sección **Justify** para alinearlo a la derecha, izquierda, centro, etcétera.

En la pestaña **Style**, cambiamos la apariencia del texto. Desde el cuadro **Global Style**, podemos elegir un **estilo global**, que agrupa distintos estilos definidos en Proteus para diferentes elementos. El estilo por defecto es el que corresponde al identificador de componente: **COMPONENT ID**. Debajo encontramos varias opciones, y a la derecha de cada una, el texto **Follow Global?**, que nos pregunta si seguiremos el estilo global, correspondiente a la selección de arriba. Si desmarcamos alguna de las opciones, habilitamos la edición o elección

de ese estilo: fuente (**Font face:**), altura (**Height:**), ancho (**Width:**), negrita (**Bold?**), itálica (**Italic?**), subrayado (**Underline?**), tachado (**Strikeout?**), visibilidad (**Visible?**) o color (**Colour:**) del texto.

Figura 6. Podemos editar las etiquetas de texto, por ejemplo, las propiedades del identificador de componente.

Ejemplo de edición de propiedades

Ahora que sabemos cómo editar las propiedades de los componentes, haremos un ejercicio, tomando otra vez el ejemplo del oscilador de la primera simulación.

Al dibujar el oscilador por primera vez, dejamos los valores por defecto del resistor y del capacitor: **R=10 kohm** y **C=1 nF**. Con estos valores, el oscilador generará una señal de aproximadamente **120 kHz**, lo cual es muy alto y no podremos apreciar en detalle.

Entonces para lograr una mejor visualización, cambiaremos los valores a **R=15 kohm** y **C=47 uF**; con esto el oscilador tendrá una frecuencia aproximada de **1 Hz** y podremos ver su funcionamiento con más claridad al correr la simulación.

Las terminales

El módulo ISIS cuenta con una función que resulta muy útil en muchos casos: **las terminales**. Con ellas podemos identificar o marcar puntos en un circuito, e incluso, hacer conexiones entre diferentes puntos sin tener que trazar líneas de conexión. Para acceder a las terminales disponibles, pulsamos el botón **Terminals Mode** en la barra de herramientas de **Modo**.

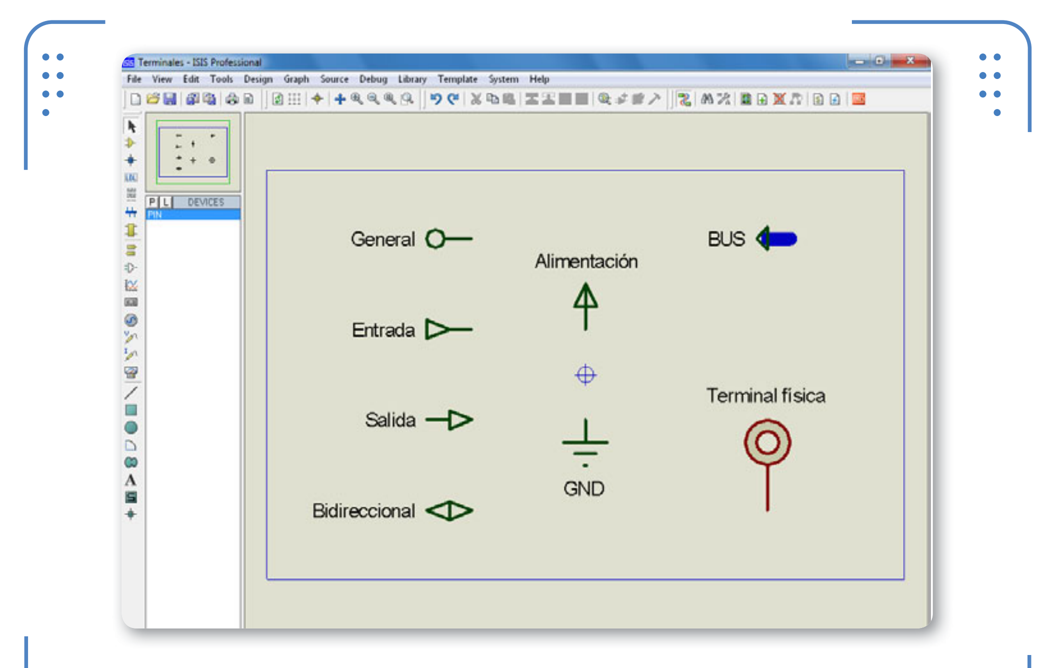


Figura 7. En el módulo ISIS es posible utilizar siete terminales lógicas y una terminal física en los diagramas.

Las cuatro primeras de la lista tienen un uso general, y es posible emplearlas para realizar desde simples referencias en los circuitos hasta conexiones entre distintos puntos. Por ejemplo, la terminal **DEFAULT** (general) puede utilizarse para marcar un punto en el circuito; **INPUT** (entrada), para definir un punto de entrada; **OUTPUT** (salida), para indicar un punto de salida; y **BIDIR** (bidireccional), para señalar un punto que puede ser de entrada o salida. El uso de estas cuatro terminales depende de la decisión de cada usuario.

Figura 8. Algunas variantes de aplicación de las diferentes terminales que podemos usar en el módulo ISIS.

Observemos como ejemplo la **Figura 8**. Las terminales **REF** marcan un punto en común, es decir, como si hubiera una línea de conexión entre los dos puntos. El mismo caso es para las terminales **CLK**, aunque en la imagen vemos una terminal de salida y otra de entrada; esto carece de importancia, ya que al tener el mismo nombre, las dos terminales indican conexión entre dos puntos.

La característica que diferencia una terminal de otra es, precisamente, su **nombre** o **etiqueta**. Es por eso que cada vez que



MODELOS DLL

Los modelos llamados **VSM DLL model** son componentes que tienen que ser programados externamente debido a su funcionamiento o a las características que deben tener en la simulación o animación; por ejemplo, un display **LM016L**. Estos modelos se agregan a Proteus mediante librerías **.DLL** que normalmente están programadas en C++.

usemos una terminal, deberemos darle una denominación. Si hacemos doble clic en cualquier terminal colocada en un diseño, se abre la ventana **Edit Terminal Label**. Como no son componentes, las terminales carecen de propiedades; solo tienen etiquetas.

Figura 9. Siempre debemos dar un nombre a las terminales para identificarlas entre las demás.

En el campo **String** tenemos que escribir el nombre para cada terminal. Este campo es una lista desplegable; por eso, si presionamos la flecha para abrir la lista, veremos los nombres de todas las demás



TERMINALES LÓGICAS Y FÍSICAS

Las terminales disponibles en el modo de terminales son solo lógicas, es decir, no representan ningún componente o elemento físico en el circuito. Si necesitamos colocar una terminal física, podemos usar el componente **PIN**, que encontramos en la ventana **Pick Devices**. Esta terminal sí tiene equivalente físico, sobre todo, al diseñar un circuito impreso, y será un **pad**.

terminales que ya tenemos en el circuito. En caso de que queramos unir dos terminales, podemos simplemente elegir un nombre de la lista en lugar de escribirlo. Las demás opciones de las pestañas **Label** y **Style** son idénticas a las de las etiquetas de los componentes.

Dos casos especiales son las terminales **GROUND** (tierra) y **POWER** (alimentación). La primera se usa como referencia o tierra, y es nombrada de forma automática como **GND** o **VSS**. La terminal de alimentación, de la misma manera, recibe la denominación **VCC/VDD**. Así, queda conectada a la red **VCC** o **VDD** y genera el voltaje correspondiente a ella, que por defecto es **5 V**. Podemos configurar las terminales **POWER** para crear nuevas líneas de alimentación o cambiar el voltaje que entregarán. En el **Capítulo 3** hablaremos en detalle sobre la configuración de las líneas de alimentación y las terminales **POWER**.

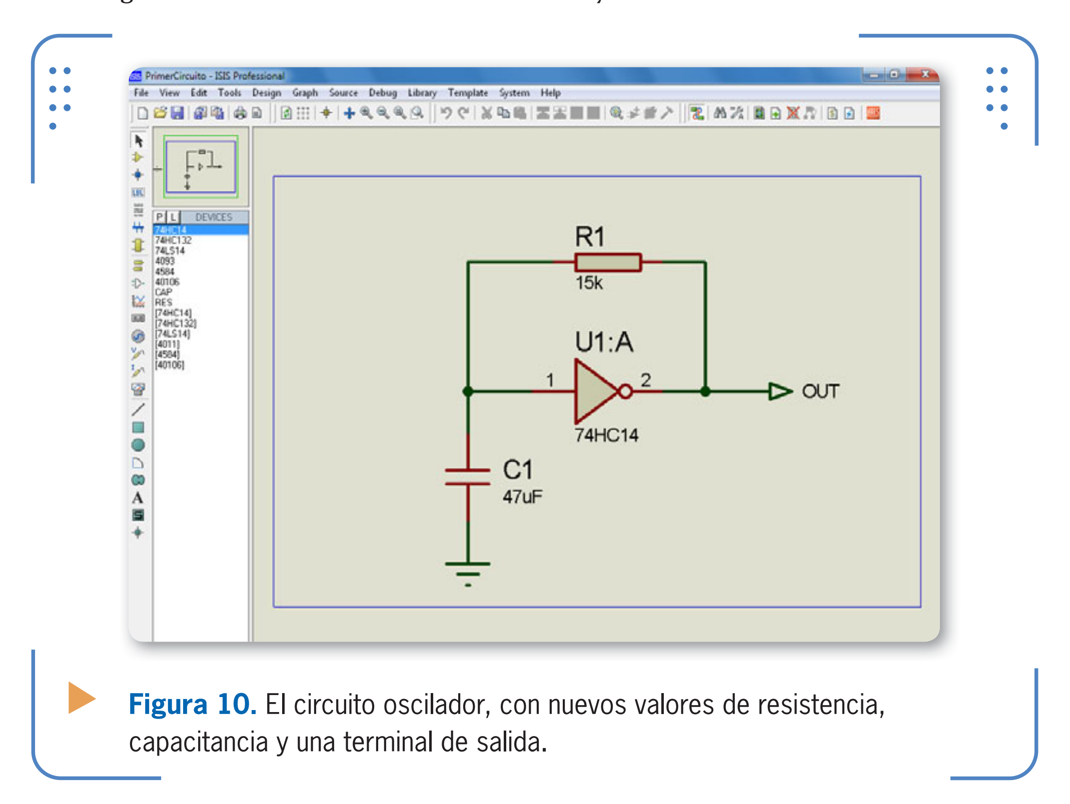


Figura 10. El circuito oscilador, con nuevos valores de resistencia, capacitancia y una terminal de salida.

También es posible colocar una terminal de salida a nuestro diagrama del oscilador para completarlo, como en la imagen que tomamos de referencia para dibujarlo.

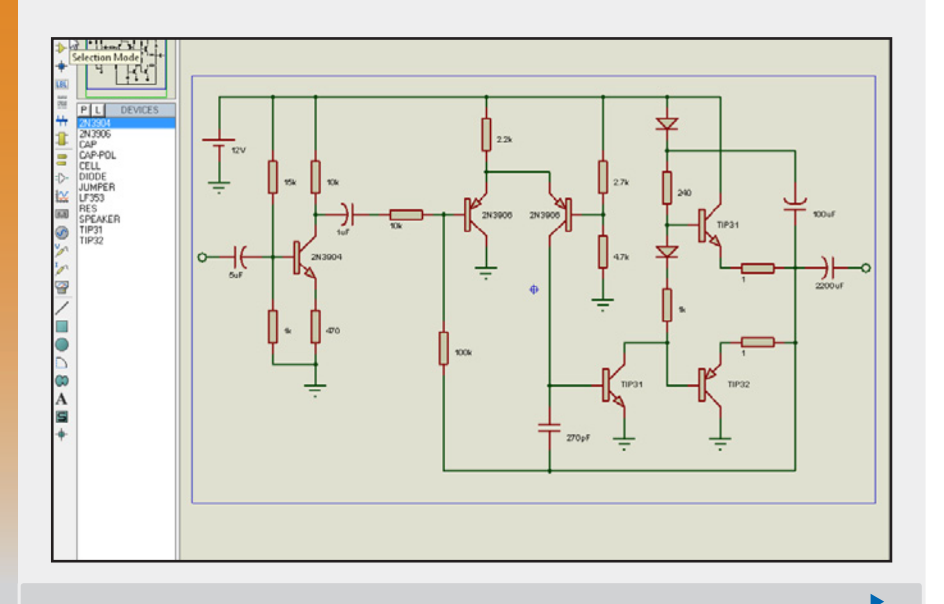
Identificar una red de conexiones

En los circuitos complejos es difícil realizar las conexiones y hay una mayor probabilidad de cometer errores. En estos casos, podemos usar la función **Highlight net on Schematic**, muy útil para identificar cómo están realizadas las conexiones y observar si son correctas. En el siguiente **Paso a paso** veremos cómo hacerlo.

PASO A PASO: RESALTAR CONEXIONES



01 En un diseño que posea líneas de conexión, presione el botón para entrar en el modo de Selección (Selection Mode).



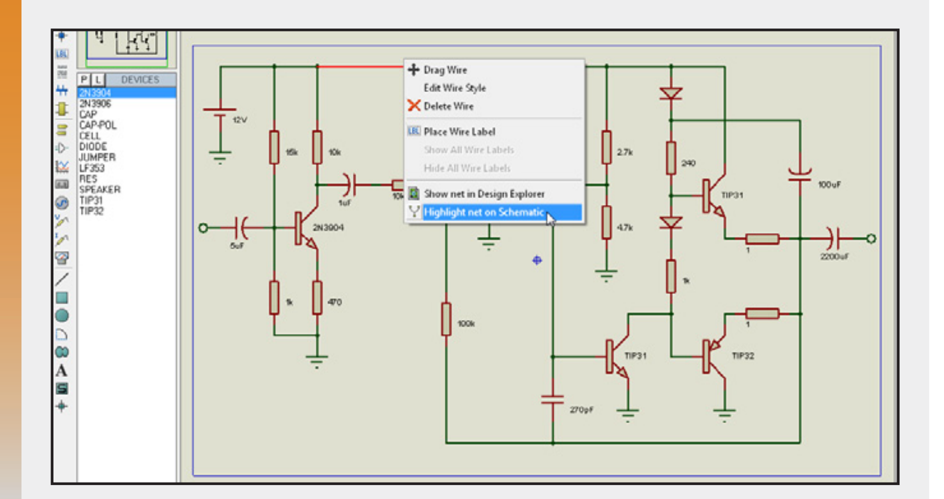


ELEGIR SOLO COMPONENTES SIMULABLES

Si pretendemos dibujar un circuito para simularlo, debemos usar solo componentes con modelo. Para esto es útil la opción **Show only parts with models?** de la ventana **Pick Devices**. Al activarla, solo aparecerán los componentes simulables, y así evitaremos colocar aquellos que no puedan ser simulados.

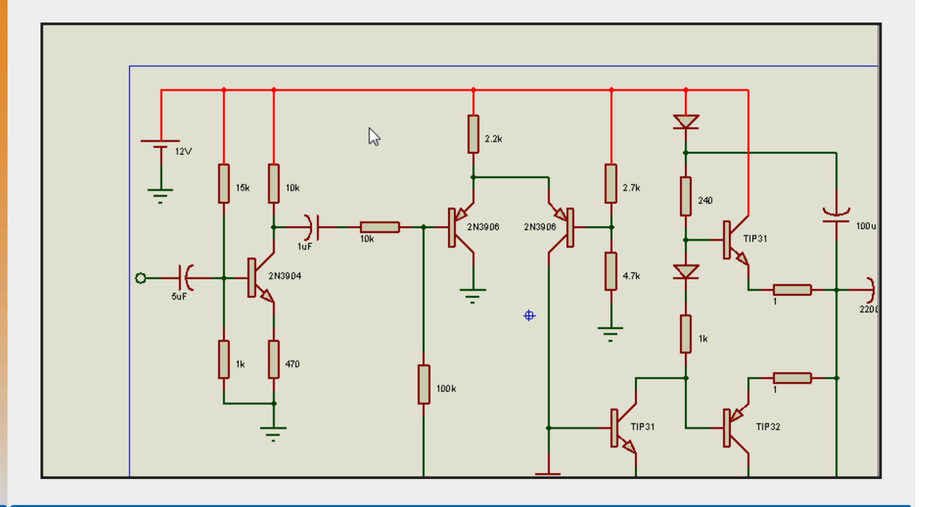
02

Haga un clic con el botón derecho sobre la línea de conexión en donde desea identificar la red de conexiones. Con esto, seleccionará el segmento de la línea y se abrirá un menú contextual.



03

En el menú, elija `Highlight net on Schematic`, la opción para resaltar la red de conexiones. A continuación, todas las conexiones que forman esta red quedarán marcadas en color rojo y, además, se centrará esa área en la ventana de `Edición`, para verla en detalle.



Con la función para identificar una red de conexiones podremos ver si las que hemos hecho están completas, si son correctas, y todos los puntos que están unidos en esa red o nodo. Esto es especialmente útil en aquellos circuitos que tienen gran tamaño.



Simulación analógica

A continuación, veremos un primer ejemplo de un circuito analógico, en el que utilizaremos un destellador simple con transistores. Además, realizaremos su simulación en Proteus.

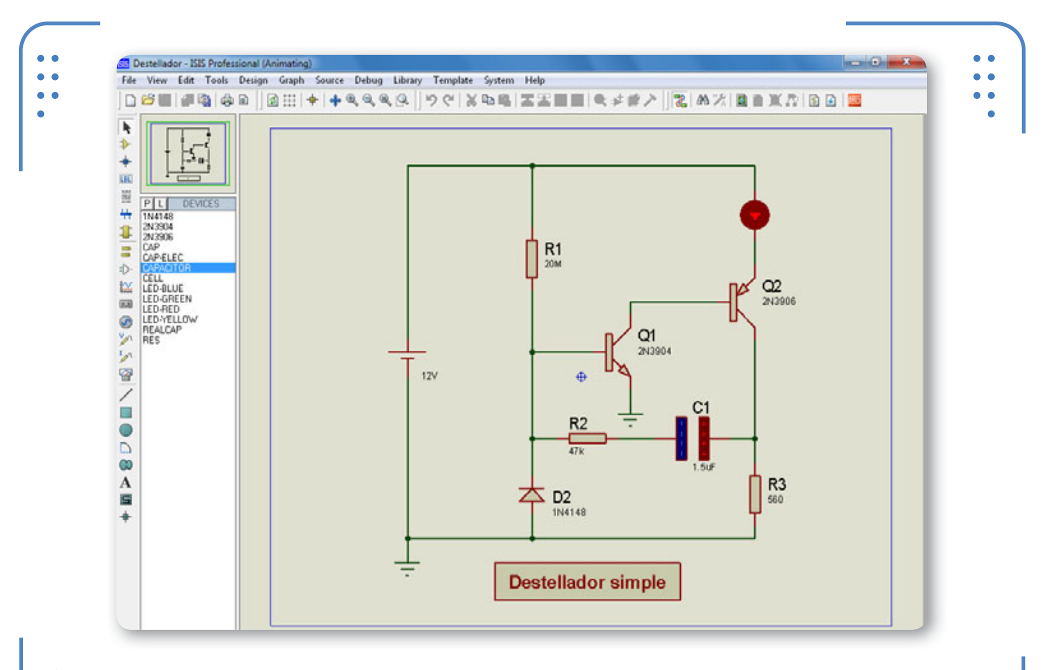


Figura 11. Tenemos un circuito destellador analógico simple, que ha sido construido con solo dos transistores.

Este circuito está diseñado para hacer destellar un LED, que puede ser útil en los automóviles como indicación falsa de una alarma activada. Es fácil de construir y requiere muy poca potencia para funcionar. Debemos ingresar en **www.redusers.com** para realizar la

descarga del archivo **Destellador.dsn**, que nos permitirá comprobar la simulación de este circuito analógico.

En este primer ejemplo hemos colocado un **capacitor animado** en el circuito, para observar su funcionamiento interno. Al correr la simulación, presionando el botón **Play** de la barra de herramientas de **Simulación**, veremos cómo el capacitor se carga y descarga durante el funcionamiento del circuito destellador, siendo parte fundamental en él.



Sondas de voltaje y de corriente

Dos herramientas sumamente útiles son las sondas de voltaje y las de corriente, que podemos usar para medir esos parámetros en puntos específicos de los circuitos. Estas sondas se encuentran en la barra de herramientas de **Modo**.

La sonda de voltaje se identifica con la letra **V**, y la sonda de corriente, con la letra **I**. Para agregar una sonda en un circuito, solo debemos seleccionarla en la barra de herramientas de **Modo** y, luego, ubicarla haciendo un clic en algún punto de la ventana de **Edición**.

Las sondas de voltaje se pueden colocar directamente en una línea de conexión del circuito, en el punto en donde necesitamos medir un voltaje, o también es posible conectarlas mediante una línea al punto de prueba, es decir, el que queremos medir.

LAS SONDAS DE VOLTAJE SE PUEDEN COLOCAR EN UNA LÍNEA DE CONEXIÓN







LOS ARCHIVOS DE EJEMPLO

Durante el desarrollo de esta obra nos referiremos a diferentes archivos de ejemplo, principalmente, con la extensión **.DSN**, que podemos descargar de la Web, desde la dirección **www.redusers.com/premium/notas_contenidos/proteus-vsm**. Así los tendremos listos para abrirlos cada vez que se haga referencia a alguno de ellos, y podremos seguir con claridad los ejemplos y proyectos estudiados.

Con las sondas de corriente hay que tener más cuidado, porque tienen una flecha. Debemos ubicar las sondas sobre la línea de conexión donde deseamos medir la corriente, y además, la flecha de la sonda tiene que ser **paralela a la línea**. Si no lo hacemos así, ISIS nos dará un error al momento de correr la simulación del circuito. No importa si la flecha está en el sentido contrario a la corriente real; en ese caso, la lectura será negativa.

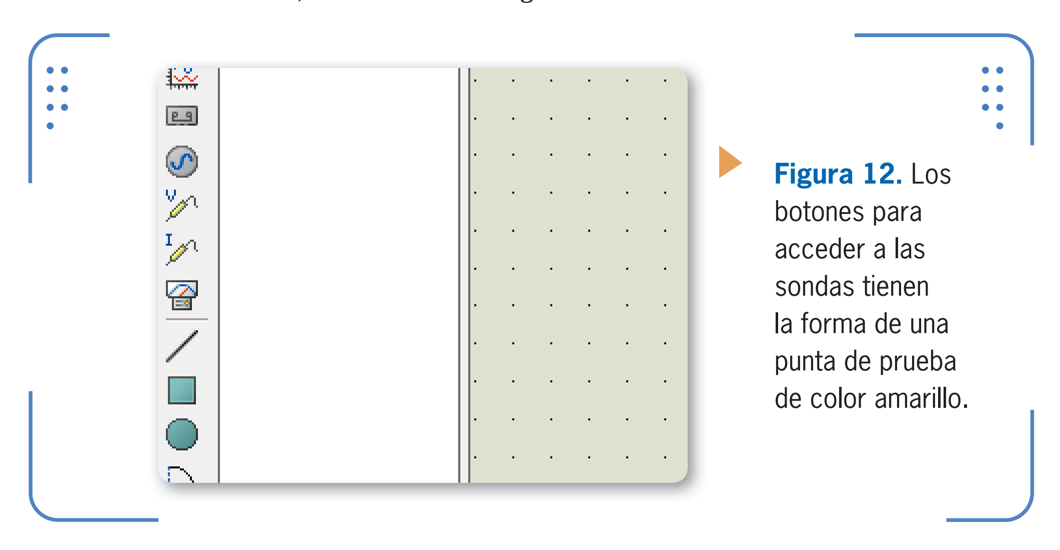


Figura 12. Los botones para acceder a las sondas tienen la forma de una punta de prueba de color amarillo.

Al colocar una sonda, ya sea de voltaje o de corriente, esta será etiquetada automáticamente con el nombre del componente más cercano al punto donde la ubiquemos en el circuito, pero podemos editar el nombre e ingresar otro diferente.

PODEMOS EDITAR EL NOMBRE POR DEFECTO DE LA SONDA E INGRESAR UNO DIFERENTE

Dentro de las propiedades de las sondas de voltaje, encontramos la opción **Load To Ground?**. Si la activamos, agregaremos una carga hacia tierra en el punto donde colocamos la sonda (como si hubiese un resistor conectado entre ese punto y la tierra) y deberemos establecer el valor de la resistencia de carga.

Esta opción puede ser de gran utilidad cuando queremos establecer una carga específica en ese punto, al mismo tiempo que usamos una sonda. Esta es otra herramienta de Proteus que nos facilita el trabajo con nuestro circuitos.

Ejemplo

Veamos un ejemplo del uso de sondas en un circuito, tomando un simple divisor de voltaje y uno de corriente. Podemos descargar el archivo **Divisores.dsn** para observar esta simulación.

Figura 13. Un sencillo divisor de voltaje y de corriente permite observar el uso de las sondas en las simulaciones.

En este ejemplo hemos agregado algunas sondas en los circuitos. Al correr la simulación, veremos que al lado de cada sonda aparecen los valores de **voltaje** y **corriente**. Observemos que las sondas de corriente (**I**, **I1**, **I2** e **Itotal**) siempre se colocan con las flechas apuntando en la dirección de la corriente, paralelas a la línea de conexión, y sobre la línea. En cambio, la sonda de voltaje (**V**) está conectada al circuito mediante una línea. Esto es perfectamente válido, pero solo para las sondas de voltaje; si hacemos esto con las de corriente, al simular obtendremos un error.

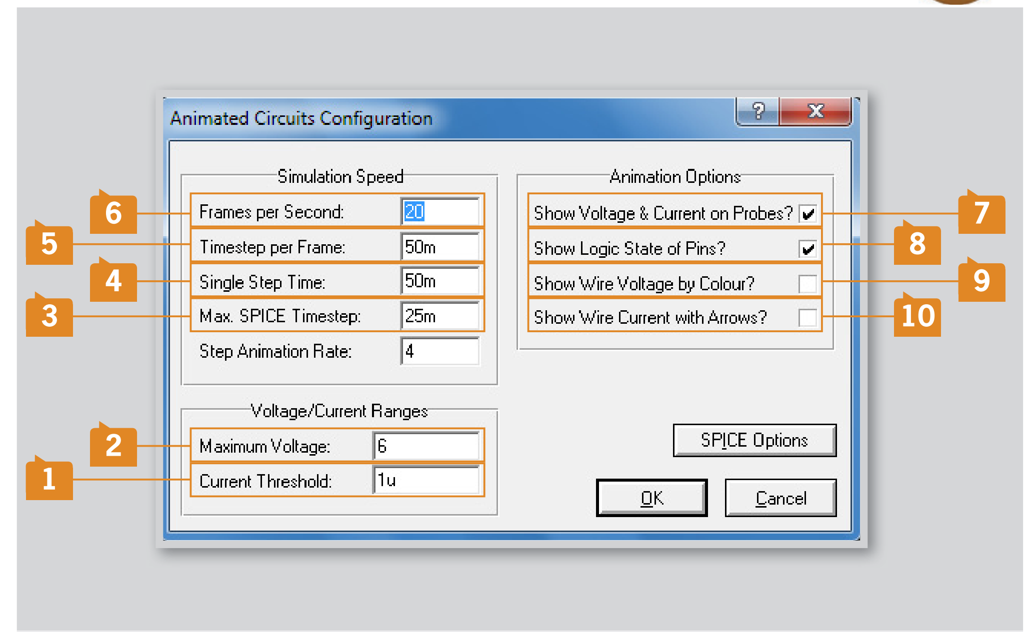
Como es fácil de apreciar, las sondas son elementos que pueden ser de gran utilidad para medir rápidamente voltajes y corrientes en diferentes puntos de un circuito.

Opciones de animación

Podemos observar mejor el funcionamiento de los circuitos cambiando algunas opciones de animación. Para hacerlo, vamos al menú **System** y seleccionamos **Set Animation Options...**. Se abrirá el cuadro de diálogo **Animated Circuits Configuration**, con dos secciones principales: **Simulation Speed** para controlar la velocidad de la simulación y **Animation options** para configurar la forma en que veremos la simulación.

GUÍA VISUAL ■ LAS OPCIONES DE ANIMACIÓN





01 **CURRENT THRESHOLD (UMBRAL DE CORRIENTE):** permite indicar el umbral de corriente. Si la corriente que circula en una línea es igual o mayor que este valor, las flechas se mostrarán en la simulación; si es menor, no.

02 **MAXIMUM VOLTAGE (VOLTAJE MÁXIMO):** para especificar el máximo voltaje en la representación de los colores en el circuito.

03 **MAX. SPICE TIMESTEP (TIEMPO MÁXIMO DE PASO DE SPICE):** fija un tiempo máximo para cada paso de simulación en el módulo SPICE. A mayor valor, la simulación será más fluida, pero puede perderse exactitud.

04	**SINGLE STEP TIME (TIEMPO DE UN PASO):** define el tiempo que transcurre con cada paso de la simulación, usando el botón **Step** (paso).
05	**TIMESTEP PER FRAME (TIEMPO POR CUADRO):** para indicar cuánto tiempo avanzará la simulación en cada cuadro de animación.
06	**FRAMES PER SECOND (CUADROS POR SEGUNDO):** configura cuántos cuadros correrán durante cada segundo de simulación; los valores deben estar entre 1 y 50.
07	**SHOW VOLTAGE & CURRENT ON PROBES? (¿MOSTRAR VOLTAJE Y CORRIENTE EN SONDAS?):** al activar esta opción, aparecerán los valores de corriente y voltaje de las sondas; si la desactivamos, las mediciones no se verán.
08	**SHOW LOGIC STATE OF PINS? (¿MOSTRAR NIVELES LÓGICOS EN LOS PINES?):** si seleccionamos esta opción, se mostrarán los niveles lógicos con pequeños cuadros de diferente color en los pines de los circuitos digitales.
09	**SHOW WIRE VOLTAGE BY COLOUR? (¿MOSTRAR VOLTAJE CON LÍNEAS DE COLORES?):** permite ver una representación de los voltajes en las líneas de conexión del circuito mediante diferentes colores.
10	**SHOW WIRE CURRENT WITH ARROWS? (¿MOSTRAR CORRIENTES EN LAS LÍNEAS CON FLECHAS?):** permite obtener una representación del sentido de la corriente en cada línea del circuito mediante flechas.

Las opciones de velocidad y de animación permiten ver la simulación de diferentes maneras, siguiendo algunas recomendaciones.

Las opciones **Frames per Second** y **Timestep per Frame** determinan la velocidad de animación del circuito; los valores por defecto son 20 y 50 m, respectivamente. Entonces, si multiplicamos 20 cuadros cada



CUIDADO CON LA VELOCIDAD

En caso de alterar la velocidad de simulación para acelerarla o hacerla más lenta, es importante no olvidarnos de regresar los valores de simulación al tiempo real o colocar una nota que informe sobre la modificación que hemos hecho, ya que esto puede confundirnos en el futuro a nosotros mismos, o a otras personas que abran la simulación y no sepan que la velocidad está alterada.

segundo, por 50 milisegundos de cada cuadro, obtenemos un tiempo exacto de **un segundo**, lo cual indica que la simulación correrá en tiempo real. Si queremos mantener esta velocidad real, debemos asegurarnos de que la multiplicación de estos campos arroje un resultado de un segundo; por ejemplo, para 50 cuadros por segundo tenemos que ingresar un tiempo por cuadro de 20 milisegundos (**20 m**).

Podemos variar estos valores para acelerar o ralentizar la simulación. Si colocamos una combinación de 20 por 100 m, por ejemplo, el resultado será 2 segundos, de modo que por cada segundo transcurrido, en la simulación habrán pasado 2 segundos, y de esta forma habremos acelerado el proceso al doble de la velocidad real.

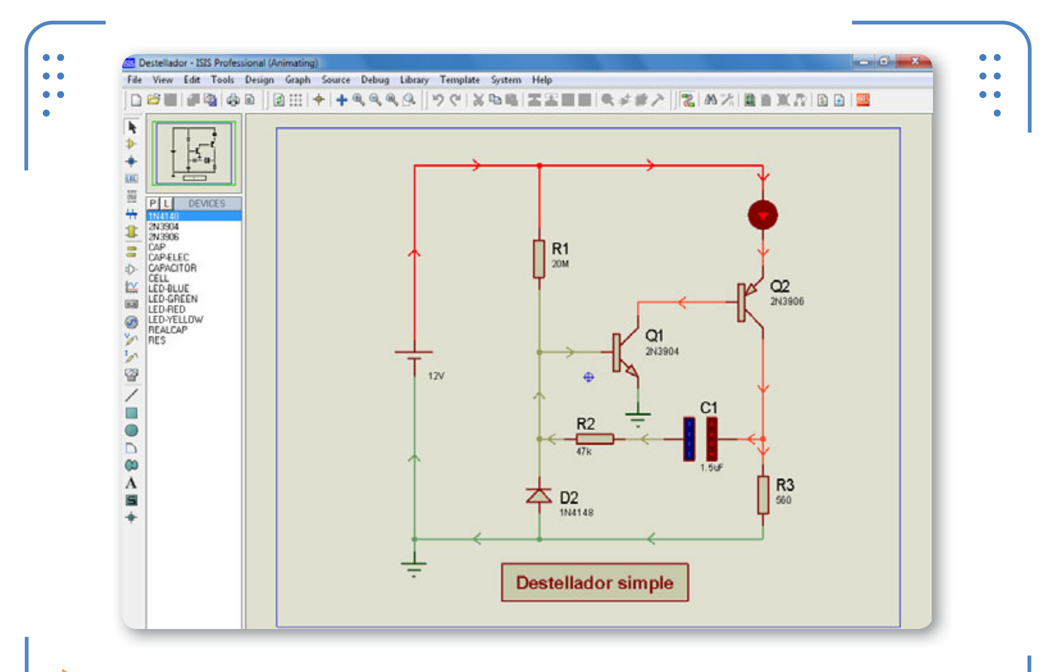


Figura 14. Las opciones de animación permiten ver el funcionamiento de un circuito en forma más detallada.

Al acelerar la simulación, los eventos pasarán más rápido y, por el contrario, al hacer más lenta la simulación, podremos ver el funcionamiento con mayor lentitud.

Para la opción **Show Voltage & Current on Probes?**, los voltajes se representan con variaciones de color de la siguiente forma: los voltajes

positivos en rojo, tierra en verde y los voltajes negativos en azul. Los niveles lógicos únicamente se muestran si el circuito es digital, con pequeños cuadros: el nivel lógico alto en rojo, el nivel lógico bajo en azul y los niveles indeterminados en gris.



Puntos de ruptura con sondas de voltaje

Las sondas de voltaje tienen una función que puede ser útil en ciertos casos: son los **puntos de ruptura en tiempo real**. Si colocamos una sonda de voltaje en un circuito y vamos a sus propiedades, veremos una sección llamada **Real Time Breakpoint**, que permite poner en pausa la simulación cuando el voltaje medido por la sonda alcanza o sobrepasa un voltaje específico.

Figura 15. Las sondas de voltaje pueden servir como puntos de ruptura analógicos o digitales.

Esta función cuenta con tres opciones: **Disabled** (deshabilitado) es la predefinida y significa que el punto de ruptura está inactivo, **Digital** se usa para un valor digital, y **Analog**, para valores analógicos. En el campo **Trigger Value** tenemos que ingresar el valor de voltaje en el cual deseamos que el punto de ruptura se dispare; para la opción **Digital** debemos colocar **1** o **0**, mientras que para **Analog** podemos establecer cualquier valor. El campo **Arm at Time** sirve para indicar el tiempo en que el punto de ruptura comenzará a estar activo. Por defecto, es 0, pero podemos establecer otro valor, por ejemplo, 2, y esto indicará que el punto de ruptura se activará solo después de 2 segundos en la simulación. La lista desplegable de la derecha brinda la posibilidad de mostrar o no un texto con la información del punto de ruptura al lado de la sonda.

Ejemplo

Utilizaremos el circuito **Destellador2.dsn**, en el que agregamos algunas sondas de voltaje y corriente. En la sonda de voltaje denominada **Vcap**, que mide el voltaje del capacitor, colocamos un punto de ruptura analógico con un voltaje de disparo de **9.5 Volts** y un tiempo de activación de 0 segundos, para que el punto de ruptura esté activo desde el momento de inicio del proceso.

EN LA BARRA DE ESTADO TENDREMOS INFORMACIÓN SOBRE EL PUNTO DE RUPTURA

Al correr la simulación, observamos que, al alcanzar el voltaje de **9.5 V** en esa sonda, la simulación se pone en pausa automáticamente. De esta manera, podemos ver los valores de voltaje y corriente que toman las otras sondas en ese momento. Esto puede ser útil precisamente para analizar diferentes parámetros del circuito cuando se alcanza un voltaje determinado en un punto específico de él.

En la barra de estado encontraremos información sobre el punto de ruptura que se ha disparado, el tiempo y el voltaje en el cual ocurrió dicho disparo. Si presionamos el botón **Play**, la simulación continuará, y el punto de ruptura actuará de nuevo al alcanzar otra vez el voltaje de disparo. Si queremos correr la simulación sin que se detenga cada vez que el punto de ruptura actúe, debemos desactivarlo dentro de las propiedades de la sonda, eligiendo **Disabled**.

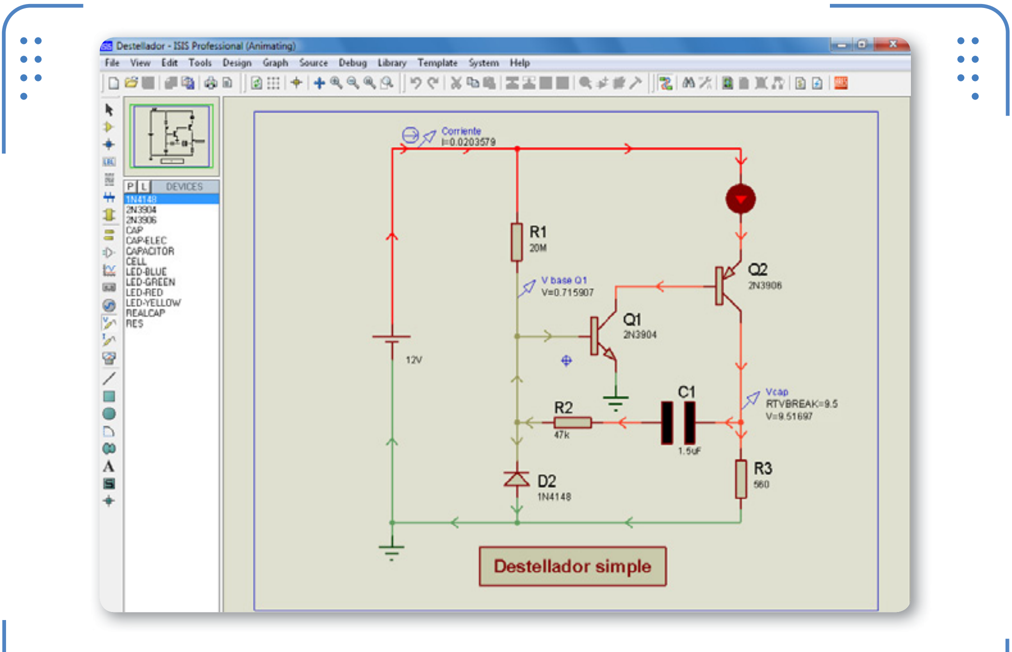


Figura 16. Podemos analizar el destellador colocando puntos de ruptura en las sondas de voltaje.



Aislar parte de un circuito con sondas de voltaje

En las propiedades de las sondas de voltaje hay una opción en la parte inferior llamada **Isolate After?**, que podemos traducir como **aislar en adelante**. Esta función sirve para aislar o desconectar lo que esté delante de la sonda. Si la marcamos, la sonda literalmente dividirá o cortará la línea de conexión en el punto donde se encuentre esta, por lo que el resto del circuito en adelante estará aislado.

En la **Figura 17**, podemos apreciar que la sonda **V** divide la línea de conexión y aísla

ISOLATE AFTER? PERMITE AISLAR LO QUE SE ENCUENTRE DELANTE DE LA SONDA



el resistor **R2**, es decir, lo desconecta del circuito. Es importante observar la posición de la sonda, ya que se aislará lo que esté delante; imaginemos a la sonda como una flecha, a partir de la cual todo se aislará. Si colocamos mal una sonda con la opción **Isolate After?**, al correr la simulación obtendremos un error, y esta no se llevará a cabo.

Es importante tomar en cuenta también que esta opción no funcionará si tenemos activado **Show Wire Current with Arrows?** dentro de las opciones de animación.

Figura 17. La opción **Isolate After?** en las sondas de voltaje corta una línea de conexión.



DESACTIVAR PUNTOS DE RUPTURA

Parece que hay un error en la versión 7.10 de Proteus, y al intentar desactivar un punto de ruptura analógico, este no se desactiva. Para hacerlo, podemos borrar la sonda y colocar otra que, por defecto, tiene los puntos de ruptura deshabilitados, o podemos pasar el punto de ruptura primero a digital y después desactivarlo; así no tendremos que borrar la sonda.



Simulación paso a paso

Aprenderemos a lograr una **simulación paso a paso**. Con el circuito en la ventana de ISIS, presionamos **Step** en la barra de herramientas de **Simulación**. El proceso comenzará y se pondrá automáticamente en **pausa** en el tiempo cero. Luego, pulsamos otra vez **Step** para avanzar otro paso, es decir, el tiempo definido en las opciones de animación en **Single Step Time**. Por ejemplo, si ingresamos un tiempo de **50 m**, la simulación avanzará hasta los 50 milisegundos de manera instantánea y se pondrá en pausa, hasta presionar de nuevo el botón **Step** para avanzar otros 50 milisegundos, cada vez que lo pulsamos.

De esta forma, podemos avanzar por lapsos de tiempo definidos. Esto es muy útil si necesitamos analizar la simulación para encontrar errores o ver los valores que toman algunos parámetros que no se perciben en una simulación continua.

PARA REALIZAR UNA SIMULACIÓN PASO A PASO DEBEMOS UTILIZAR EL BOTÓN STEP



Es posible definir cualquier tiempo para los pasos en las opciones de animación, desde nanosegundos hasta horas, aunque siempre dado en **segundos**. Por ejemplo, para un minuto debemos colocar 60 segundos; para 10 minutos, 600 segundos, etcétera. Es importante tener en cuenta que una simulación paso a paso con valores altos de tiempo por paso no simulará en tiempo real. Por ejemplo, si colocamos 100 como tiempo de un paso, al presionar **Step** la simulación avanzará 100 segundos de manera instantánea y no deberemos esperar ese tiempo.

Como en cada paso la simulación quedará en pausa, también podemos presionar el botón **Play** para que avance de forma continua en cualquier momento. Podemos alternar el uso de los botones **Play** y **Step**.





CUIDADO AL AISLAR

Debemos tener cuidado con la opción **Isolate After?**. Si la activamos y la dejamos así al guardar nuestro circuito, en el futuro puede confundirnos. Es posible que el circuito se comporte de manera extraña o no funcione debido a que hemos aislado una parte con alguna sonda de voltaje y no lo recordamos.



Carga del procesador al simular

Aunque podemos simular prácticamente cualquier circuito analógico en Proteus, la desventaja es que este tipo de simulación requiere muchos **recursos de procesamiento** en la computadora. La simulación de circuitos se hace mediante modelos matemáticos del comportamiento de cada componente, y a partir de ahí se calcula el comportamiento del circuito completo. Esto implica una enorme cantidad de cálculos matemáticos por segundo y, además, demanda un porcentaje de procesamiento extra para la animación del circuito.

> LA SIMULACIÓN ANALÓGICA REQUIERE MUCHOS RECURSOS DE PROCESAMIENTO

De esta forma, cuanto más complejo sea un circuito, más recursos de procesamiento necesitará al momento de correr la simulación. Como vimos anteriormente, durante una simulación, en la barra de estado aparece el tiempo transcurrido, y entre paréntesis, la carga del microprocesador de la computadora (**CPU Load %**). Esto representa cuán exigido se encuentra el procesador durante la simulación: mientras más alto sea este porcentaje, mayor será la carga; incluso, puede llegar al 100% en circuitos muy complejos. Si la carga del procesador es superior al 70% o 75%, la simulación no podrá correr en **tiempo real** y lo hará más lentamente.



Simulación digital

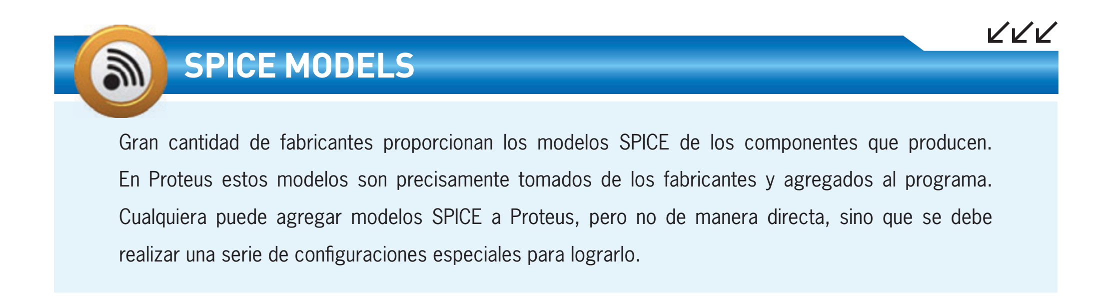
La simulación digital es más eficiente que la analógica. Aunque un circuito puramente digital sea muy complejo, de todos modos será bien simulado, sin usar muchos recursos de procesamiento y corriendo en tiempo real. Como podemos imaginar, Proteus posee algunos componentes o herramientas que permiten simular los circuitos digitales de forma eficiente, rápida y fácil.

Sondas lógicas y estados lógicos

Dos herramientas que serán de suma utilidad son los **estados lógicos** y las **sondas lógicas**. Podemos encontrarlas en la categoría **Debugging Tools** en la ventana **Pick Devices**, o también buscarlas por sus nombres: **LOGICSTATE**, **LOGICPROBE** o **LOGICTOGGLE**.

Figura 18. Las sondas lógicas analizan los niveles digitales, y los estados lógicos introducen niveles en el circuito.

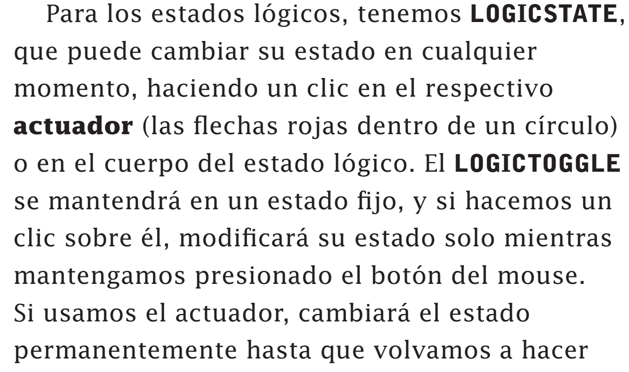
Los estados lógicos nos permiten definir un nivel lógico en alguna parte de nuestro circuito, por ejemplo, en una entrada de una


SPICE MODELS

Gran cantidad de fabricantes proporcionan los modelos SPICE de los componentes que producen. En Proteus estos modelos son precisamente tomados de los fabricantes y agregados al programa. Cualquiera puede agregar modelos SPICE a Proteus, pero no de manera directa, sino que se debe realizar una serie de configuraciones especiales para lograrlo.

compuerta lógica. Mientras tanto, las sondas nos muestran el nivel lógico en algún punto, por ejemplo, en la salida de una compuerta. En las sondas lógicas se leerá el nivel actual como **0**, **1**, o **?**, si hay un nivel indefinido. Las sondas tienen dos tamaños que podemos elegir: pequeño o grande, para una mejor visualización.

> LAS SONDAS LÓGICAS MUESTRAN EL NIVEL LÓGICO EN ALGÚN PUNTO DEL CIRCUITO



Para los estados lógicos, tenemos **LOGICSTATE**, que puede cambiar su estado en cualquier momento, haciendo un clic en el respectivo **actuador** (las flechas rojas dentro de un círculo) o en el cuerpo del estado lógico. El **LOGICTOGGLE** se mantendrá en un estado fijo, y si hacemos un clic sobre él, modificará su estado solo mientras mantengamos presionado el botón del mouse. Si usamos el actuador, cambiará el estado permanentemente hasta que volvamos a hacer un clic sobre él o en el cuerpo del estado lógico. Si deseamos cambiar el estado inicial, es decir, el estado permanente, debemos entrar en sus propiedades y elegir **Low** o **High** en la lista **Initial State**. Los estados y sondas lógicas se colocan y conectan como cualquier otro componente en el dibujo del circuito.

Ejemplo

Ahora veamos un ejemplo del uso de los estados lógicos y las sondas lógicas en un circuito. Para esto debemos descargar el archivo **BCDa7seg.dsn**, que posee un circuito integrado **4543**, el cual es un decodificador de BCD a 7 segmentos y encenderá un display con el valor que hay en sus entradas en BCD.

Hemos colocado estados lógicos en las entradas **A**, **B**, **C** y **D** para poder dar un valor en BCD al 4543 con ellos. También agregamos un estado lógico momentáneo en la entrada **Latch Enable** (**LE**); de esta forma, después de cambiar el valor de los estados lógicos en **A**, **B**, **C**, **D**, debemos presionar el estado colocado en **LE** para dar un pulso y hacer que el nuevo valor se refleje en las salidas del 4543 y en el display. En las salidas **QA** a **QG**, ubicamos sondas lógicas para ver su nivel lógico en todo momento, independientemente de que se reflejen en el display. De esta forma, podemos apreciar el funcionamiento y la utilidad de los estados lógicos y las sondas lógicas en los circuitos digitales.

Figura 19. Un controlador de display es un buen ejemplo para estudiar las sondas y los estados lógicos.



Simulación mixta

La simulación mixta es la que involucra una parte digital y otra analógica. Proteus es capaz de simular circuitos mixtos sin ningún problema, aunque debemos considerar la complejidad del circuito si necesitamos hacer una simulación en tiempo real.

Proyecto: efecto de luces

Veamos un ejemplo de una simulación mixta para la cual utilizaremos el archivo **EfectoLuces.dsn**.

Este circuito es muy simple, ya que solo requiere un circuito integrado para lograr un efecto de luces desplazándose con LEDs. Está construido a partir de un inversor con disparador Schmitt; en este caso usamos el circuito integrado **74HC14**.

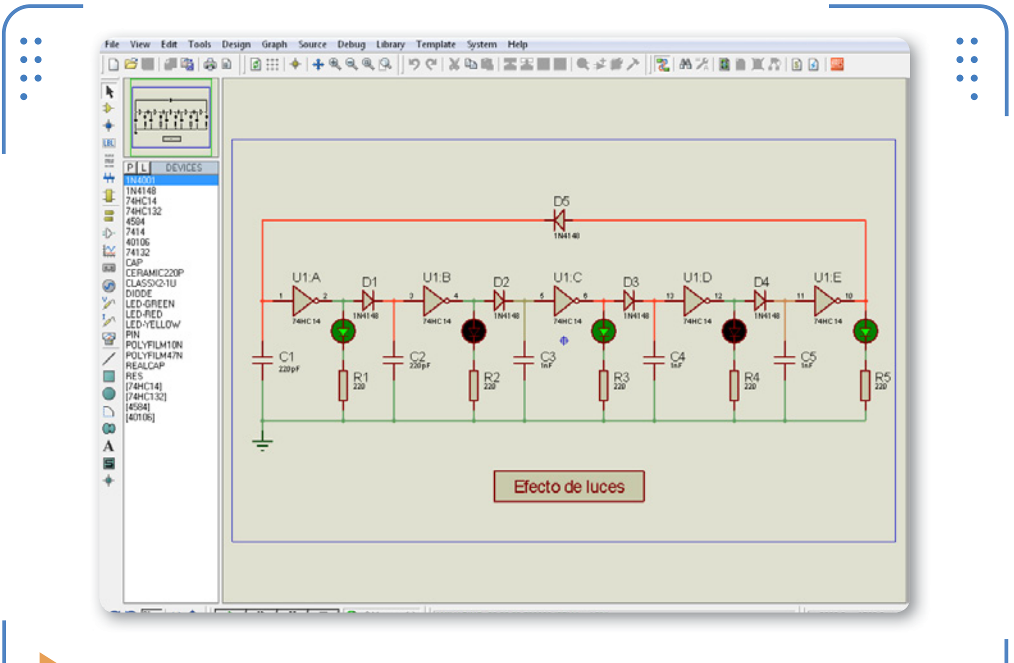


Figura 20. Un efecto de luces con LEDs es un interesante ejemplo de un circuito mixto de tipo sencillo.

Es mixto porque el 74HC14 es un circuito digital, pero los diodos 1N4148 y los capacitores son componentes analógicos, y de ellos depende en gran medida el correcto funcionamiento de este circuito.

LOS LEDS ANALÓGICOS CONLLEVAN MAYOR CARGA EN LA SIMULACIÓN



Al abrir el archivo **EfectoLuces.dsn** y correr la simulación, veremos en funcionamiento un circuito mixto en Proteus.

Notaremos dos detalles importantes en él. Los **capacitores** son algo especiales. Además de los específicos y genéricos, existe un modelo llamado **REALCAP**, que es un tipo de capacitor real, en donde es posible especificar no solo la capacitancia sino también otros parámetros. En la **Figura 21** vemos la ventana de propiedades del REALCAP. Los parámetros que se pueden manejar son: resistencia de fuga (**Leak resistance**), ESR (**Equivalent Series Resistance**), inductancia serie (**Series Inductance**), resistencia de aislamiento (**Shunt Resistence**) y carga inicial (**Initial Charge**) en Volts.

Este tipo de capacitor es útil en simulaciones en las que se requiere tomar en cuenta estos parámetros. Si se usan capacitores genéricos, la simulación no funcionará bien o puede no funcionar, ya que los capacitores son esenciales para este circuito. En este ejemplo hemos dejado los valores por defecto de todos los parámetros, ya que son valores estándar para la mayoría de los capacitores; excepto en el capacitor **C1**, en el que colocamos una carga inicial de 5 Volts para que el circuito arranque rápidamente.

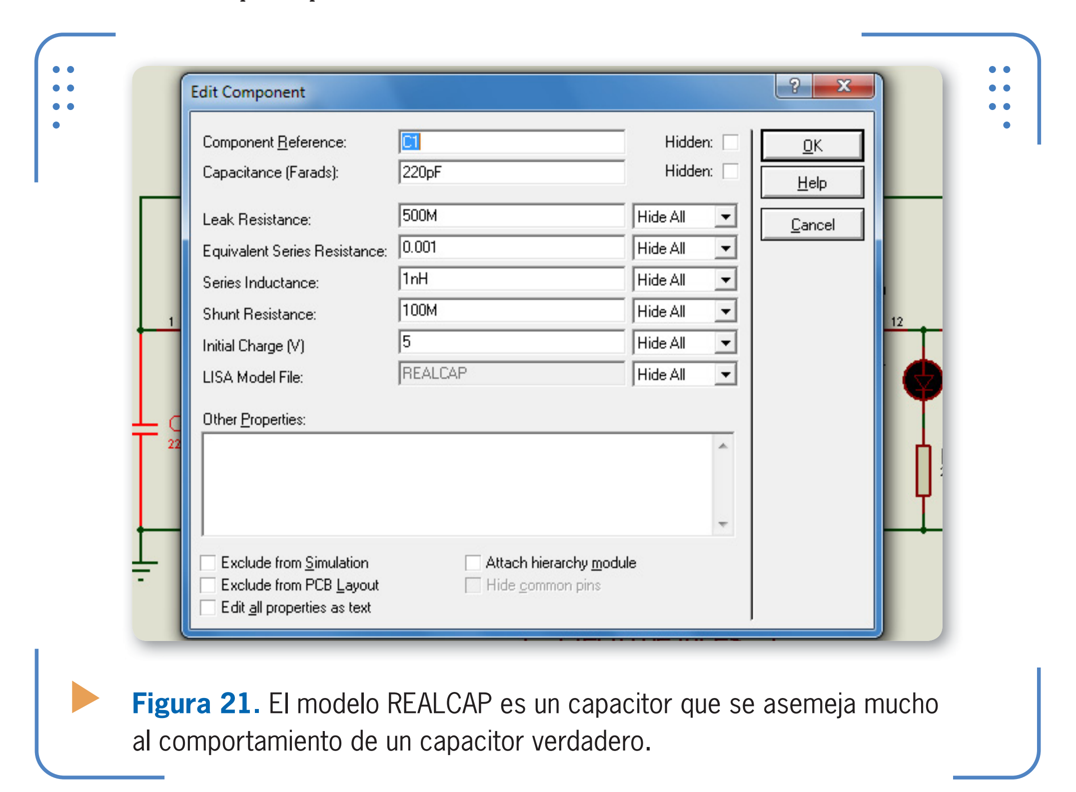


Figura 21. El modelo REALCAP es un capacitor que se asemeja mucho al comportamiento de un capacitor verdadero.

Si accedemos a las propiedades de los **LEDs**, veremos que se puede elegir entre varios parámetros, y entre ellos, un modelo **analógico** o **digital** (**Model type**). Un LED digital hará las simulaciones más ligeras, ya que se toma al LED solo como un indicador de estados, es decir: encendido o apagado. Por su parte, en un LED analógico se consideran los parámetros reales, y esto conlleva mayor carga al momento de simular, especialmente, si tenemos un circuito complejo o con muchos LEDs. En este ejemplo hemos configurado los LEDs como digitales.



Visualización de parámetros instantáneos

Durante cualquier simulación podemos presionar el botón **Pause** (el botón **Step** también funcionará) para colocarla en estado de pausa. Mientras la simulación permanezca en este estado, llevaremos el cursor del mouse sobre cualquier componente y este tomará la forma de una mano. Al hacer un clic sobre cualquier componente del circuito, se mostrará una ventana que contiene información sobre los parámetros del componente en ese momento.

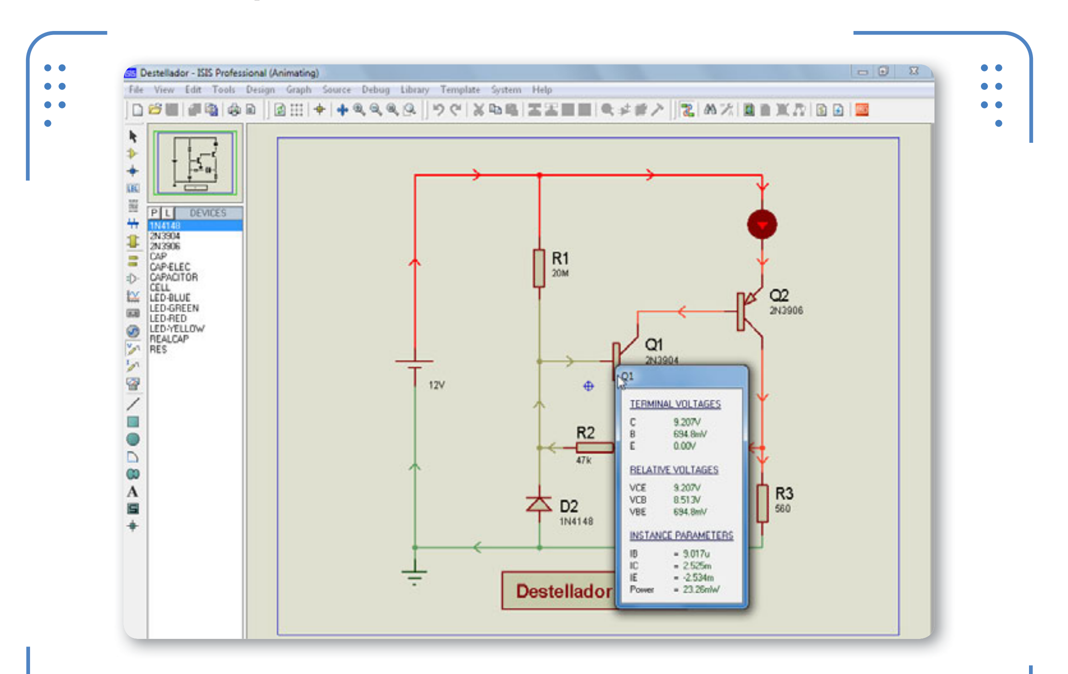


Figura 22. Podemos tener información útil de cada componente durante una pausa en las simulaciones.

En la **Figura 22** vemos un ejemplo con el destellador. Colocamos pausa en el tiempo de **4 segundos** (el tiempo exacto se logra simulando paso a paso) y, al hacer un clic en el transistor **Q1**, se mostrarán sus parámetros instantáneos:

- Los voltajes en los pines (**TERMINAL VOLTAGES**)
- Los voltajes relativos entre los pines (**RELATIVE VOLTAGES**)
- Los parámetros de corriente y potencia (**INSTANCE PARAMETERS**)

Estos valores se toman directamente, es decir, se miden justo en el momento de la pausa; en este ejemplo, a los 4 segundos. Otros elementos presentarán diferentes parámetros según el componente de que se trate. Los circuitos integrados mostrarán el voltaje o los niveles lógicos en todos sus pines, dependiendo de si es analógico o digital. Al mover el cursor del mouse, la pequeña ventana desaparecerá y podremos hacer clic en otro componente, si lo deseamos.

En algunos componentes, sobre todo en los que tienen **actuadores**, no se puede acceder a esta función con un clic, ya que es como si presionáramos los actuadores. En estos casos, hacemos un clic con el botón derecho del mouse, y en el menú contextual seleccionamos la opción **Operating Point Info**.



RESUMEN

En este capítulo hemos concluido con las opciones básicas para dibujar diagramas de circuitos electrónicos en ISIS. Además, comenzamos a simular circuitos en Proteus, y aprendimos cómo configurar las opciones de animación y de velocidad. Ya tenemos las bases de la simulación, pero esto es solo el principio. También estudiamos algunos componentes especiales y herramientas útiles para nuestras simulaciones. En el capítulo siguiente, analizaremos los generadores como herramientas para la inyección de señales en los circuitos.

Actividades

TEST DE AUTOEVALUACIÓN

1. ¿Qué componentes contiene la categoría **Analog ICs**?
2. ¿Para qué se utiliza el botón **Play**?
3. ¿Cómo se accede al reporte de simulación?
4. ¿Cómo se coloca una terminal?
5. Mencione tres componentes pertenecientes a la librería ACTIVE.
6. ¿Para qué sirve una sonda de voltaje?
7. ¿Cómo se coloca correctamente una sonda de corriente?
8. ¿Qué opción debemos activar en las propiedades de animación para ver con colores los voltajes en las líneas de conexión?
9. ¿Para qué sirve un estado lógico (LOGICSTATE)?
10. ¿Cómo se inicia una simulación paso a paso?

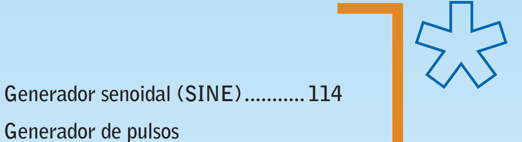
ACTIVIDADES PRÁCTICAS

1. Abra la ventana **Pick Devices**, en el campo de búsqueda escriba la palabra **active**, y navegue por los resultados para familiarizarse con los componentes de esta librería.
2. Abra el circuito **Destellador.dsn** y cambie el capacitor animado por un **capacitor genérico** del mismo valor para comprobar que el funcionamiento es el mismo.
3. Descargue el circuito **LED_analogico-digital.dsn**, en el cual se ha agregado un LED configurado como analógico a la salida del circuito oscilador con 74HC14; observe su funcionamiento al simular.
4. Cambie la configuración del LED en el circuito **LED_analogico-digital.dsn** por un modelo digital. Observe los cambios al momento de simular.
5. Busque un circuito sencillo, dibújelo en ISIS e intente simularlo.



Líneas de alimentación y generadores

En muchos de los circuitos electrónicos que nos interese simular, será importante poder generar señales para alimentarlos o estimularlos. En Proteus existen diferentes maneras de generar todo tipo de señales, ya sean analógicas o digitales. En este capítulo aprenderemos cómo ISIS provee de alimentación a los circuitos y, además, veremos la posibilidad de obtener señales mediante los generadores.



Servicio de atención al lector: **usershop@redusers.com**



Pines ocultos en circuitos integrados

En el circuito del efecto de luces del **Capítulo 2**, notaremos un detalle particular: los cinco inversores pertenecen al circuito integrado **74HC14**, pero en el diagrama se muestran como elementos individuales. Muchos de los CI en ISIS se presentan así, principalmente, las compuertas lógicas, los amplificadores operacionales o los flip flops. Si observamos los identificadores, veremos que el nombre del primer inversor es **U1:A**; esto identifica al circuito integrado número **1**, y la **A**, al primer inversor de este CI. Como sabemos, el 74HC14 contiene seis inversores, así que los del primer CI estarán identificados con una letra de la **A** a la **F**. Si agregamos más inversores al diseño, el séptimo se identificará como **U2:A**, lo que significa que pertenece a un nuevo circuito integrado **U2**.

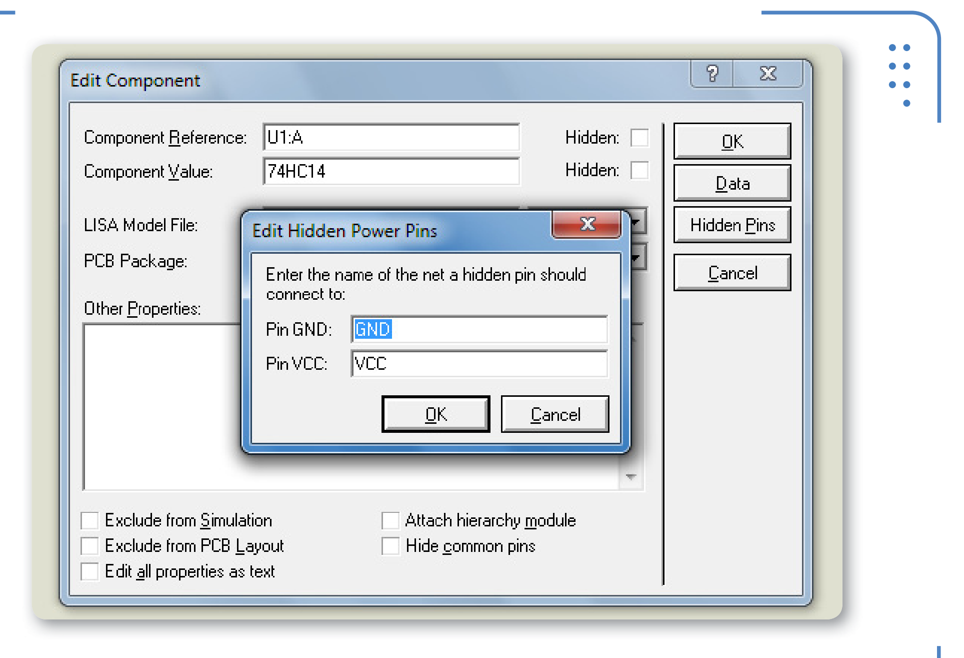


Figura 1. Podemos ver o cambiar los nombres de los pines ocultos con el botón **Hidden Pins**.

Otra característica es que no podemos ver los pines de alimentación del circuito integrado 74HC14, y que carecemos de fuente de voltaje o batería. Esto es porque en ISIS muchos de los circuitos integrados tienen los pines de alimentación ocultos para simplificar los diagramas, principalmente, los digitales. En la ventana de propiedades del 74HC14 (de cualquier compuerta individual), encontraremos en la parte derecha un botón con el nombre **Hidden Pins** (pines ocultos). Si lo presionamos, se abre la ventana **Edit Hidden Power Pins** (editar pines de alimentación ocultos).

En esta ventana leeremos lo siguiente: **Enter the name of the net a hidden should connect to**, que en español significa **ingrese el nombre de la red a la cual un pin oculto debe conectarse**. Esto nos permite seleccionar la red a donde se conectarán los pines ocultos. En este caso, tenemos el pin GND o tierra, que se conectará a la red GND. Como sabemos, al colocar una terminal de tierra, esta automáticamente se llamará GND o VSS, así que este pin oculto del 74HC14 se conectará a las terminales de tierra. Por su parte, el pin VCC se conectará a la red VCC, que provee un voltaje de 5 V de corriente directa.

MUCHOS CIRCUITOS INTEGRADOS TIENEN LOS PINES DE ALIMENTACIÓN OCULTOS



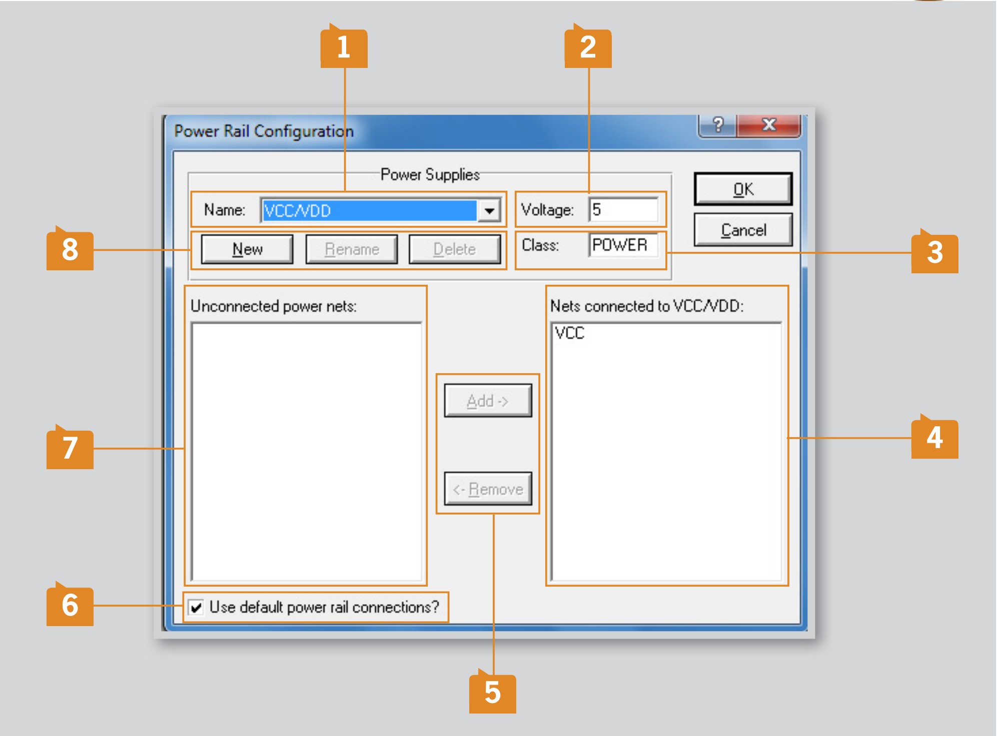


Configuración de líneas de alimentación

El módulo ISIS tiene la capacidad de generar de forma automática los **voltajes de alimentación de corriente directa** para las simulaciones, especialmente, en el caso de los circuitos integrados digitales. Si necesitamos ver o modificar la configuración de estas líneas de alimentación, tenemos que ingresar al menú **Design** y seleccionar la opción **Configure Power rails**. A continuación, se abrirá la ventana **Power rail configuration** (configuración de líneas de alimentación), que nos permitirá modificar diferentes parámetros de estas líneas. En la siguiente **Guía visual** explicaremos cada una de estas opciones de manera detallada.

GUÍA VISUAL ■ LÍNEAS DE ALIMENTACIÓN





01 **NAME (NOMBRE):** en esta lista tenemos los nombres de las líneas o fuentes de alimentación disponibles.

02 **VOLTAGE (VOLTAJE):** aquí podemos ver o cambiar el valor de voltaje que provee la línea de alimentación elegida en el campo **Name**.

03 **CLASS (CLASE):** establece la clase; por defecto, todas serán **POWER**, y normalmente no necesitaremos cambiar este campo.

04 **NETS CONNECTED TO (REDES CONECTADAS A):** en este recuadro figuran las redes que ya están conectadas a la fuente de alimentación definida en el campo **Name**.

05 **ADD, REMOVE (AGREGAR, REMOVER):** estos botones permiten agregar o quitar redes para conectarlas o desconectarlas de las líneas de alimentación.

06 **USE DEFAULT POWER RAIL CONNECTIONS? (¿USAR CONEXIONES DE LÍNEAS DE ALIMENTACIÓN POR DEFECTO?):** permite cargar las opciones predefinidas de la configuración. Podemos dejarla marcada y realizar cambios, pero si la desmarcamos y luego la seleccionamos otra vez, se nos pedirá confirmación para cargar los valores por defecto. Al hacerlo, se borrarán las configuraciones personalizadas.

07 **UNCONNECTED POWER NETS (REDES DE ALIMENTACIÓN NO CONECTADAS):** en este recuadro aparecen las redes de alimentación que no están conectadas a ninguna fuente de alimentación.

08 **NEW, RENAME, DELETE (NUEVO, RENOMBRAR, BORRAR):** estos botones nos permiten crear una nueva línea o fuente de alimentación, renombrarla o borrarla.

Al crear un nuevo documento o diseño en ISIS, la configuración de las líneas o fuentes de alimentación tendrá valores por defecto.

En el campo **Name**, encontraremos tres líneas de alimentación: **VCC/VDD** con un voltaje por defecto de **5 V**, la línea **GND** con un voltaje de **0 V** y la línea **VEE** con **-5 V**.

Para la línea GND no es posible cambiar ningún tipo de parámetro, a excepción de la clase, ya que, de manera predefinida, GND o tierra siempre tendrá un valor de 0 Volts. En cambio, para las líneas VCC/VDD y VEE solo podremos modificar el valor del voltaje y la clase.

Además, debemos saber que no está permitido renombrar ni borrar estas fuentes de alimentación.

DE MANERA PREDEFINIDA, GND O TIERRA SIEMPRE TENDRÁ UN VALOR DE 0 VOLTS





ELEMENTOS Y PINES OCULTOS

Hemos visto con anterioridad que algunos circuitos integrados presentan elementos individuales en el diagrama, como en el caso de las compuertas lógicas. Sin embargo, si necesitamos cambiar el nombre de los pines ocultos en ellos, debemos hacerlo en todos los elementos; por ejemplo, en el 74HC14 tendremos que hacerlo en cada uno de los inversores.

Líneas de alimentación y terminales POWER

Para crear nuevas líneas de alimentación, es posible utilizar las terminales **POWER**. Como hemos visto, estas terminales por defecto se conectan a la línea **VCC/VDD**, a menos que cambiemos su nombre. Para crear una línea o fuente de alimentación con una terminal POWER, podemos cambiar su nombre ingresando directamente el valor de voltaje deseado, pero siempre debemos comenzar con un signo más (**+**) o menos (**-**), ya que con esto especificamos la polaridad del nuevo voltaje. Por ejemplo, si necesitamos crear una línea de alimentación de **12 Volts**, colocaremos una terminal POWER llamada **+12** o **+12 V**.

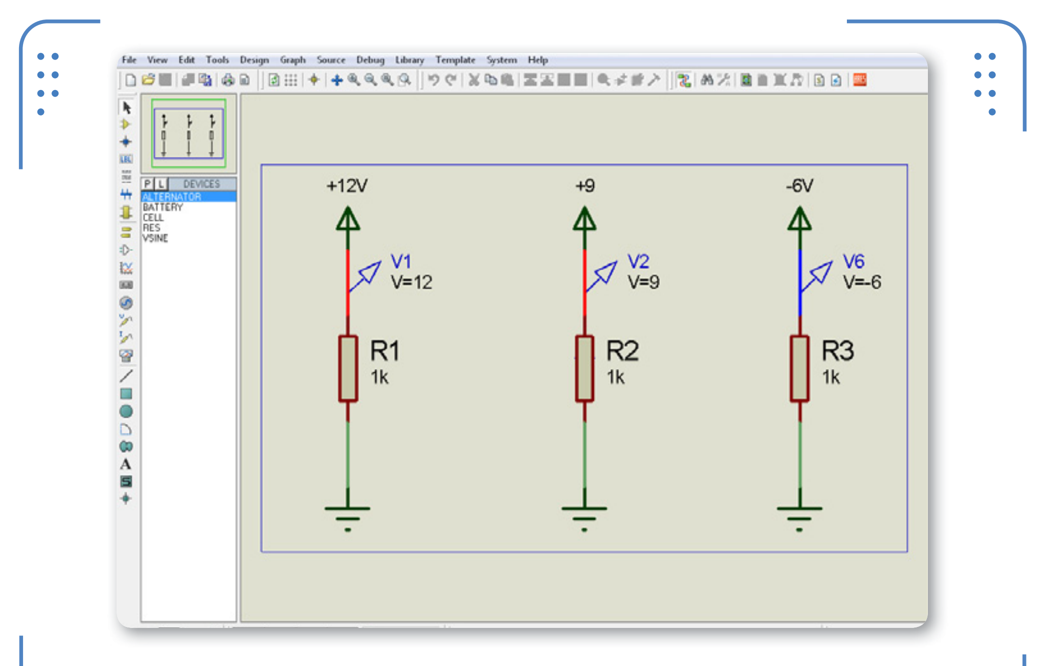


Figura 2. Diferentes ejemplos de líneas de alimentación con valores de voltaje ingresados de manera directa.

En la **Figura 2** vemos ejemplos de líneas de alimentación que utilizan terminales POWER, con un valor de voltaje ingresado directamente como nombre. Si vamos a la ventana de configuración de estas líneas, notaremos que en el campo **Name** aparecen las nuevas

líneas o fuentes en la lista. Los valores del campo **Voltage** están en gris y no podemos modificarlos, ya que están definidos por el propio nombre de la terminal POWER. Si ingresamos un valor sin signo en el nombre, se tomará como texto, por eso es importante incluir el signo.

También es posible crear nuevas líneas de alimentación utilizando **nombres** en lugar de valores de voltaje. Al agregar una terminal POWER, podemos colocarle un nombre, por ejemplo **V1**, pero debemos asignarle una fuente de voltaje nueva o conectarla a una ya existente en la ventana de configuración. Si no lo hacemos, en el momento en que intentemos realizar la simulación del circuito, obtendremos un error y esta no podrá iniciar.

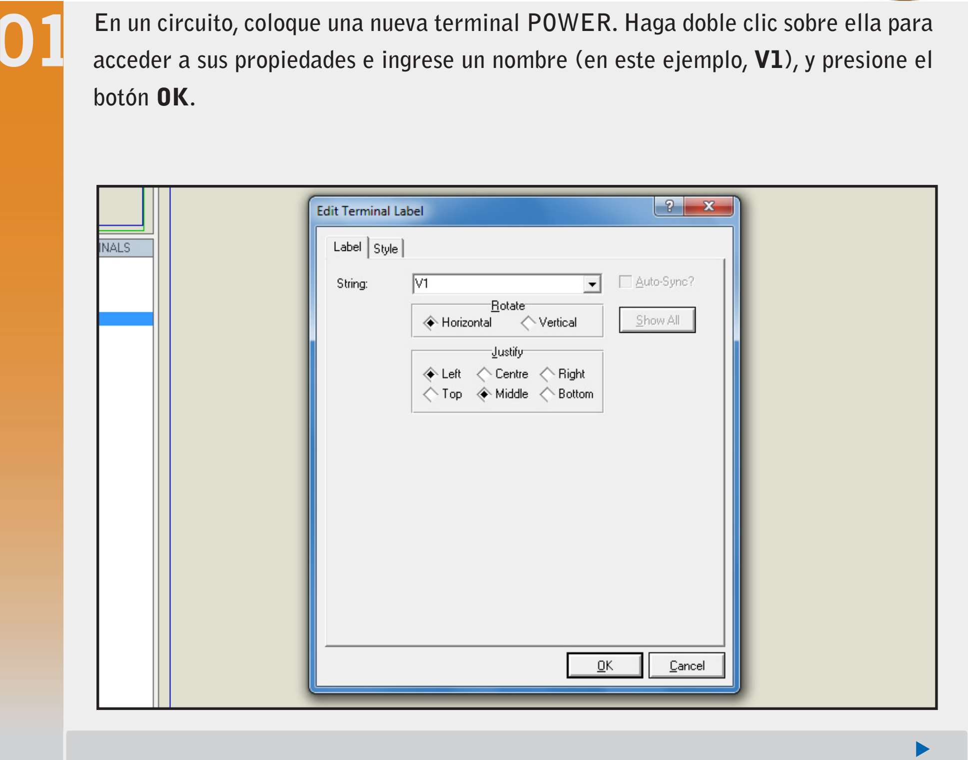
PARA CREAR LÍNEAS DE ALIMENTACIÓN PODEMOS UTILIZAR LAS TERMINALES POWER



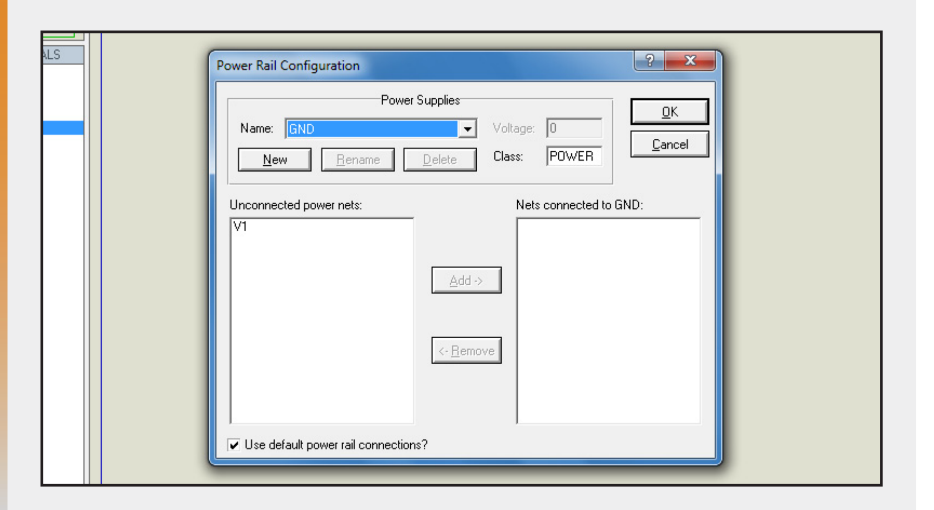
PASO A PASO: CREAR LÍNEAS CON NOMBRES



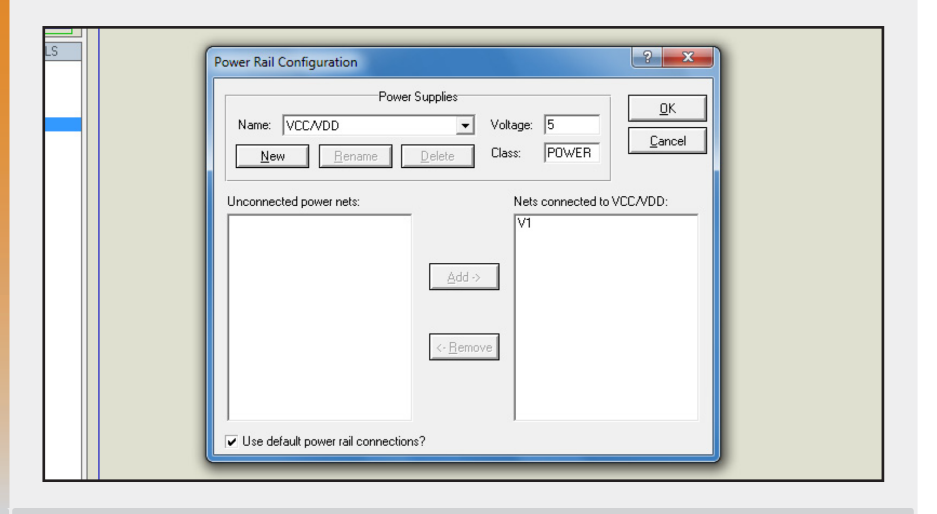
01 En un circuito, coloque una nueva terminal POWER. Haga doble clic sobre ella para acceder a sus propiedades e ingrese un nombre (en este ejemplo, **V1**), y presione el botón **OK**.



02 Vaya al menú `Design/Configure Power Rails...` y, en la ventana de configuración, observe que la nueva fuente aparece en la lista `Unconnected power nets`, pero no en la lista Name.

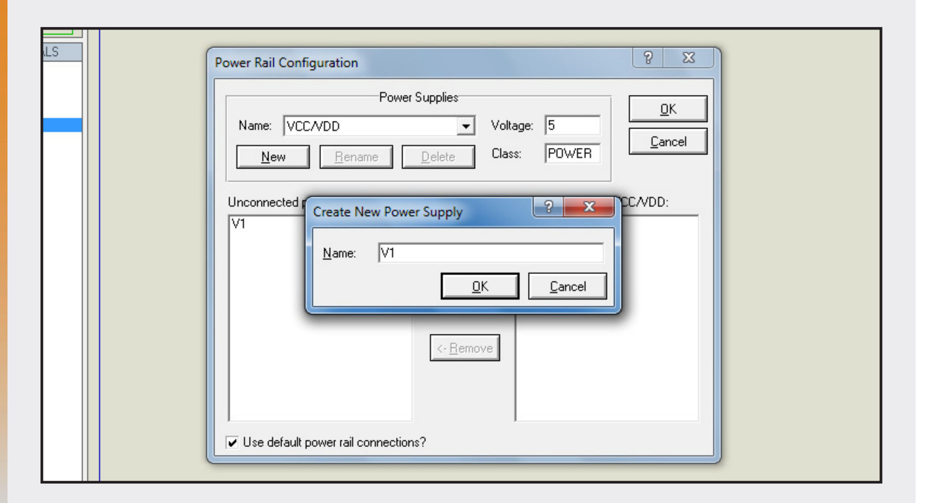


03 Tiene dos opciones. Conectar la nueva fuente a una existente, como VCC/VDD, para lo cual debe seleccionar VCC/VDD de la lista Name, elegir V1 en `Unconnected power nets` y presionar Add para agregarla a `Nets connected to VCC/VDD`. Si necesita un voltaje diferente, revierta este paso seleccionando V1 y Remove.



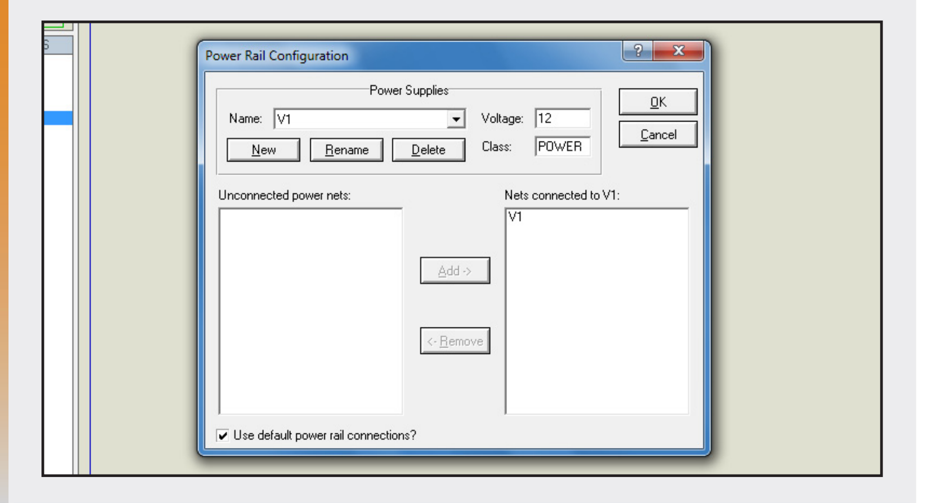


La otra opción es crear una línea de alimentación nueva presionando New y, en la ventana `Create New Power Supply`, ingresar el nombre de la línea, que puede ser igual o diferente del nombre de la terminal **POWER**. En este caso será igual, V1, de modo que presione OK. La nueva línea de alimentación aparecerá en la lista Name.



05

Para concluir, ingrese un valor de voltaje en el campo `Voltage`, por ejemplo, 12; luego seleccione V1 en la lista `Unconnected power nets` y presione Add para asignar esta línea a la fuente V1 en la lista `Nets connected to V1`. Finalmente, pulse OK y ya tendrá su nueva línea de alimentación configurada a **12 V**.



En el caso de que queramos conectar una nueva terminal POWER a una fuente existente, en sus propiedades podemos elegirla de la lista desplegable del campo denonimado **String**, donde aparecen todas las fuentes que se encuentran disponibles.

Ahora ya sabemos cómo generar nuevas líneas de alimentación utilizando terminales POWER, tanto con valores de voltaje como con nombres. Estamos listos para continuar nuestro aprendizaje.

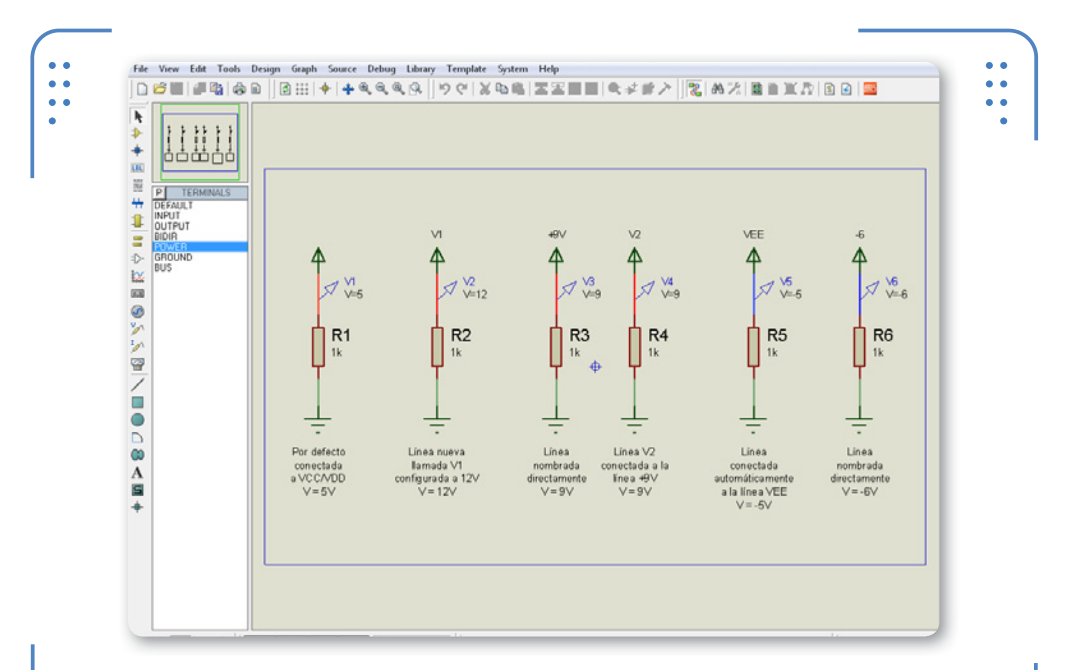


Figura 3. Diferentes ejemplos de la configuración de múltiples líneas de alimentación en el módulo ISIS.



USO DE BATERÍAS

Podemos usar las fuentes o líneas de alimentación para dar los voltajes de corriente directa durante las simulaciones de los circuitos, aunque debemos saber que también es posible utilizar diferentes baterías (**BATTERY**). Si usamos la batería que pertenece a la librería **ACTIVE**, podemos definir la resistencia interna (**Internal Resistance**) dentro de sus propiedades.



Los generadores de señales

En muchas de las simulaciones, además de los voltajes de alimentación, necesitaremos otros tipos de señales. En estos casos, ISIS cuenta con herramientas para la generación de señales digitales o analógicas, nos referimos a los **generadores**. Para acceder a los diferentes tipos disponibles, pulsamos el botón **Generator Mode** (un círculo gris con una señal senoidal azul) en la barra de herramientas de **Modo**. Entraremos en modo **Generador**, y en el **Selector de objetos** aparecerá una lista con todos los existentes.

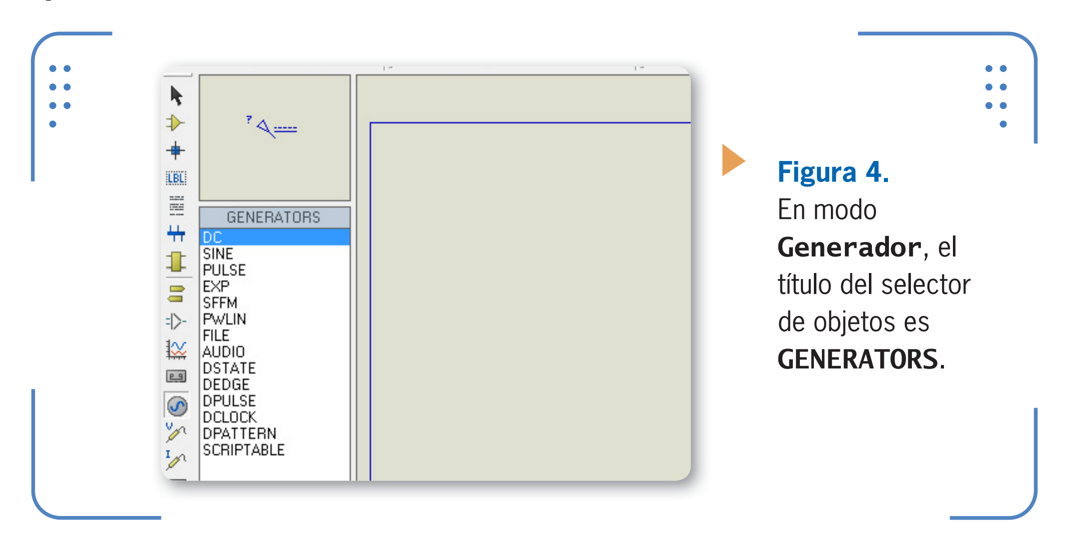


Figura 4. En modo **Generador**, el título del selector de objetos es **GENERATORS**.

Los generadores tienen la forma de un triángulo azul con una punta, y cada uno se acompaña de un símbolo que representa el tipo de señal que producirá. Para colocar un generador en un circuito, solo debemos

FUENTES DEPENDIENTES

Si necesitamos usar fuentes dependientes en nuestro proyecto, por supuesto que Proteus también cuenta con ellas. Podemos encontrarlas en la categoría **Modelling Primitives** dentro de la subcategoría **Analog (SPICE)**. Disponemos, por ejemplo, de una fuente de voltaje controlada por voltaje **VCVS** y de una fuente de corriente controlada por voltaje **VCCS**, entre muchas otras.

entrar en modo **Generador**, seleccionar uno de la lista en el S**elector de objetos** y ubicarlo en el circuito. Se puede poner el nuevo generador sobre una línea de conexión o en un espacio vacío, y luego conectarlo a un pin o a una línea de conexión existente.

Generador de corriente directa (DC)

El primer generador que encontramos en la lista es el **DC** (**Direct Current**), que produce un voltaje de corriente directa. Si colocamos un generador de esta clase en un circuito y hacemos doble clic sobre él, podemos editar sus propiedades.

Figura 5. Las propiedades del generador DC nos permiten establecer sus parámetros y otras características.

En la **Figura 5**, vemos la ventana **DC Generator Properties**, dividida en dos por una línea vertical; del lado izquierdo hay una sección que es igual para todos los generadores. Lo primero que encontramos es el campo **Generator Name**, donde ingresaremos el nombre del generador; después tenemos las secciones **Analog types** y **Digital types**, con una

lista de todos los generadores disponibles, analógicos y digitales, respectivamente. Podemos cambiar el tipo de generador desde esta ventana de propiedades, simplemente eligiendo el que necesitemos.

En el lado derecho de la ventana, vemos la configuración de los parámetros del generador, que dependerá del tipo seleccionado. Para el DC, solo tenemos el campo **Voltage (Volts)**, en el cual elegiremos el voltaje deseado para él.

Por último, en la parte inferior izquierda de la ventana encontramos cuatro opciones de configuración adicionales. A continuación vamos a explicar en detalle para qué sirve cada una de ellas:

- **Current source?**: permite convertir el generador en una fuente de corriente en vez de en una fuente de voltaje. Si marcamos esta opción en las propiedades del generador DC, entonces veremos que el campo **Voltage (Volts)** cambiará a **Current (Amps)** para definir la corriente que suministrará el generador. Esta posibilidad solo se encuentra disponible en los generadores analógicos.
- **Isolate Before?**: sirve para aislar lo que se encuentre conectado antes del generador. Esto es útil si intentamos reemplazar la señal producida por un circuito por otra con un generador. Para hacerlo, colocamos el generador directamente sobre una línea de conexión, ya que este dividirá la línea para aislar lo que esté antes de él.
- **Manual Edits?**: hará que los parámetros o propiedades del generador del lado derecho puedan ser editados como texto.
- **Hide Properties?**: brinda la posibilidad de mostrar u ocultar las propiedades del generador en el diagrama.

EL GENERADOR SE PUEDE PONER EN UN ESPACIO VACÍO Y LUEGO CONECTARLO







NOMBRES EN GENERADORES

Los generadores toman un nombre automáticamente al colocarlos o conectarlos a una red. Si cambiamos manualmente el nombre de algún generador, este quedará fijo, incluso si lo desconectamos o conectamos a otra red. Si dejamos en blanco el campo de nombre y presionamos **OK**, el generador regresará a la función de nombre automático.

Proyecto: termómetro con LEDs

Veamos un ejemplo del uso de generadores DC. Para esto, simularemos un termómetro de LEDs, con el archivo **Termometro.dsn**.

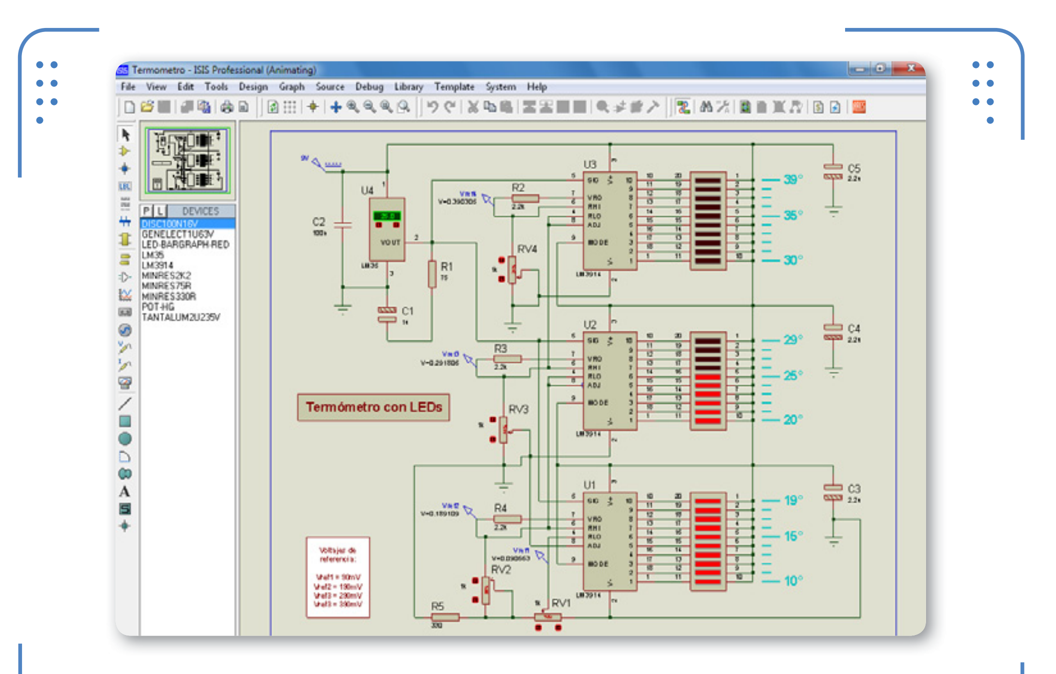


Figura 6. Un termómetro de LEDs nos servirá como ejemplo del uso de generadores de corriente directa.

Este circuito requiere un sensor de temperatura **LM35**, y como elementos para mostrar la temperatura se utilizan tres circuitos **LM3914** en cascada y **30 LEDs** o **tres barras de LEDs** para obtener un rango de lectura de 10 a 39 grados centígrados. Hemos usado tres barras de LEDs, que podemos encontrar en la categoría **Optoelectronics/ Bargraph displays** o buscando el nombre **LED-BARGRAPH**.

En el cuerpo del símbolo del sensor LM35, tenemos una **lectura numérica** y un par de **actuadores**, con los cuales podemos controlar la temperatura que medirá el sensor en la simulación; de esta forma, es sencillo simular este tipo de sensores en Proteus.

En esta simulación hemos empleado un generador DC como fuente de alimentación, que vemos en la parte superior izquierda, marcado como **9 V**. Ninguno de los circuitos integrados que usamos tiene pines

ocultos, por lo que este generador es el que alimenta a todo el circuito.

De esta manera, a partir de un ejemplo muy sencillo, es posible apreciar la forma en que los generadores de corriente directa (DC) pueden actuar como fuentes de alimentación en los circuitos.

Para el correcto funcionamiento del circuito es preciso ajustar los **voltajes de referencia** marcados por las sondas de voltaje. En el mismo diagrama tenemos una tabla con los valores de estos voltajes, que deben ser ajustados con los potenciómetros **RV1**, **RV2**, **RV3** y **RV4**. Al correr la simulación, veremos que es algo difícil ajustar los valores de los voltajes de referencia, aunque en la realidad podemos usar potenciómetros multivueltas para hacerlo con exactitud. En la simulación tal vez resulte difícil lograr valores exactos, de modo que podemos ayudarnos con otros generadores DC.

EN ESTA SIMULACIÓN HEMOS UTILIZADO UN GENERADOR DC COMO FUENTE DE ALIMENTACIÓN



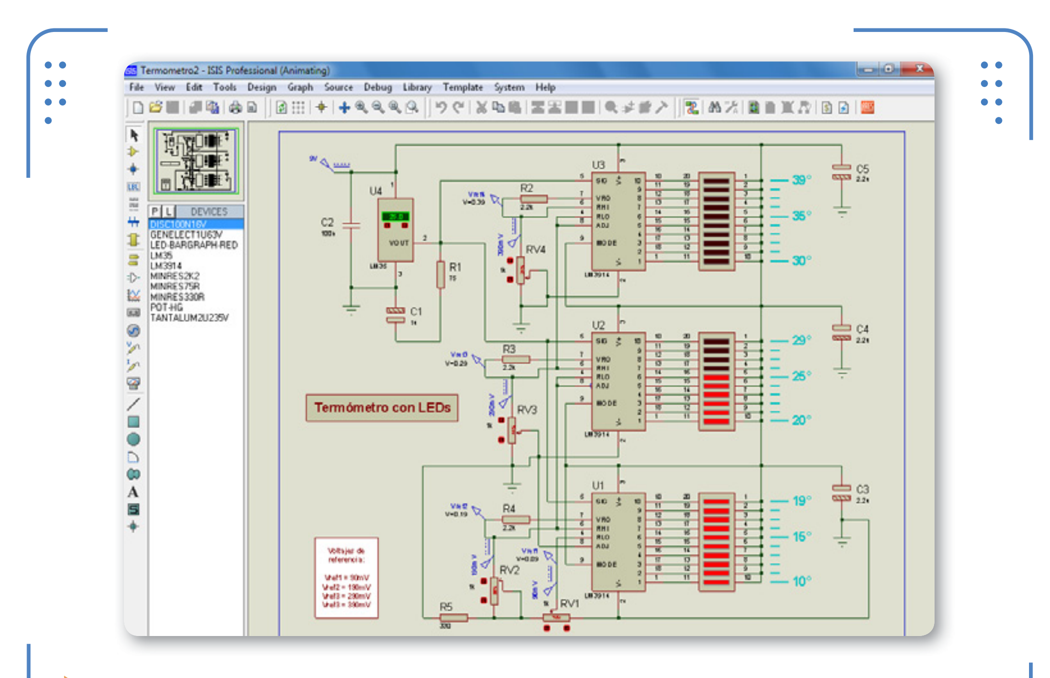


Figura 7. Podemos sustituir los potenciómetros por generadores DC para dar los voltajes de referencia exactos.

Si descargamos y abrimos el archivo **Termometro2.dsn**, veremos que se han agregado generadores de corriente directa en los puntos donde se necesitan los voltajes de referencia, cada uno generando el voltaje exacto, y además, con la opción **Isolate before?** activada para desconectar los potenciómetros sin necesidad de borrarlos.

ES IMPORTANTE LA ORIENTACIÓN DE LOS GENERADORES PARA AISLAR CORRECTAMENTE



Es muy importante tener en cuenta la orientación de los generadores para aislar correctamente y evitar errores durante la simulación. De esta manera, tenemos valores exactos en los voltajes de referencia, lo que nos permite observar el funcionamiento del circuito de una forma más precisa.

Aunque en este circuito los valores no son críticos, en otros proyectos quizá sí necesitemos voltajes exactos, y entonces nos auxiliaremos con los generadores.

Para construir este circuito debemos tener en cuenta algunas recomendaciones: los resistores R1 a R5 deben tener el 1% de tolerancia, los capacitores C3 a C5 tienen que ser de tantalio y, como ya mencionamos, los potenciómetros tienen que ser multivueltas para poder ajustar los voltajes de referencia a un valor que sea lo más exacto posible. Si necesitamos ampliar el rango de medición, tendremos que agregar más circuitos LM3914.

Generador senoidal (SINE)

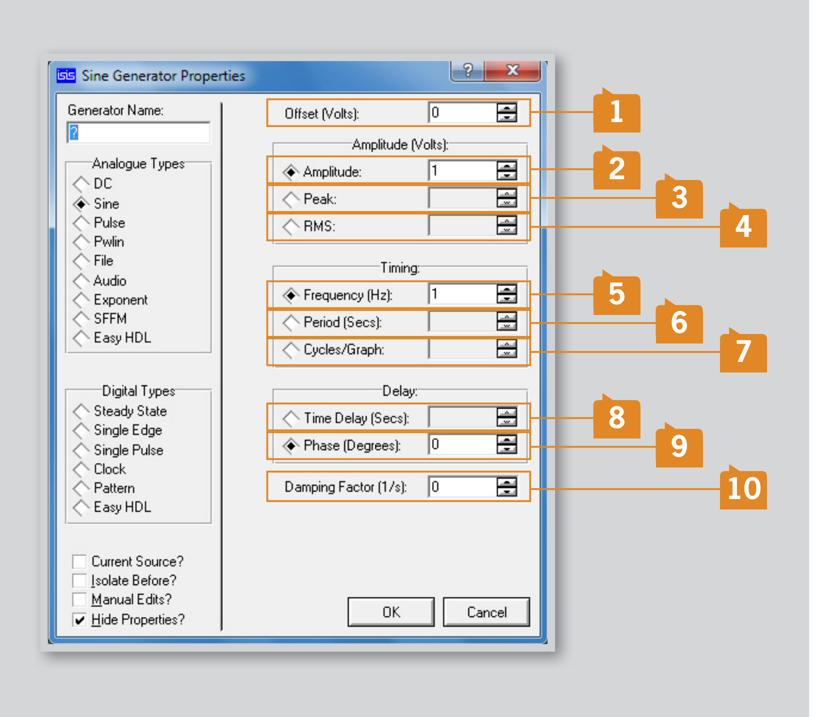
El segundo generador que encontramos en la lista es el **SINE**, que puede generar una señal **senoidal**, dados sus diferentes parámetros. En la siguiente **Guía visual**, veremos cómo ajustarlos.

AJUSTAR RANGO DE MEDICIÓN EN EL LM35

Si editamos como texto las propiedades del sensor LM35 utilizado en la simulación del termómetro, veremos que aparecen un par de ellas denominadas **MIN** y **MAX**, cuya función es limitar el rango de temperatura que medirá el sensor. En este caso, las hemos definido en **MIN=9** y **MAX=39**. Estas opciones no figuran en el modo normal.

▼ GUÍA VISUAL ■ CONFIGURACIÓN DE SINE





01 **OFFSET (VOLTS) (VOLTAJE OFFSET):** define el voltaje offset, también conocido como voltaje de desplazamiento. Es el valor en Volts al cual la señal senoidal estará desplazada verticalmente. Se suele decir que la señal está montada sobre un voltaje de CD.

02 **AMPLITUDE (AMPLITUD):** en la sección **Amplitude (Volts)** tenemos las opciones para configurar la amplitud de la señal; esta primera es para definir la amplitud o voltaje pico en Volts.

03 **PEAK (VOLTAJE PICO A PICO):** permite establecer la amplitud de la señal expresada en un voltaje pico a pico.

04 **RMS (VOLTAJE RMS):** en este espacio podemos ingresar un valor expresado en Volts RMS.

05 **FREQUENCY (HZ) (FRECUENCIA):** es el primer campo de la sección **Timing** (temporización) y define la frecuencia de la señal expresada en Hertz. ▶

06 **PERIOD (SECS) (PERÍODO):** aquí se puede especificar el período de la señal, es decir, el tiempo que dura un ciclo, en segundos.

07 **CYCLES/GRAPH (CICLOS POR GRÁFICO):** permite generar un número determinado de ciclos de la señal, esto depende de un gráfico de simulación. Hablaremos sobre este tema en el **Capítulo 5**.

08 **TIME DELAY (SECS) (TIEMPO DE RETARDO):** en este campo se puede definir un tiempo de retardo para el inicio de la señal, es decir que la señal comenzará a generarse al pasar este tiempo, establecido en segundos.

09 **PHASE (DEGREES) (FASE):** al seleccionar esta opción, podemos definir la fase de la señal en grados.

10 **DAMPING FACTOR (1/S) (FACTOR DE AMORTIGUAMIENTO):** si necesitamos una señal senoidal amortiguada, podemos establecer en este campo el factor de amortiguación.

Generador de pulsos analógicos (PULSE)

El generador **PULSE** es el siguiente que encontramos en la lista. Se utiliza para formar pulsos analógicos; podemos ver un ejemplo de este tipo de señales en la **Figura 8**.

Con esta clase de generador, es posible generar un pulso de un valor de voltaje deseado, por ejemplo, puede ir de los 2 V a los 12 V, y esto es lo que lo hace diferente de un pulso digital.

Es importante tener en cuenta que los tiempos de subida y bajada para este generador nunca pueden ser 0. Esto se debe a las limitaciones internas del simulador analógico ProSPICE. Por lo tanto, obtener una



TRIANGULAR Y DIENTE DE SIERRA

El generador PULSE puede ser útil para producir señales triangulares o de diente de sierra, simplemente definiendo un ancho de pulso igual a 0 segundos. Para una señal triangular debemos colocar iguales tiempos de bajada y subida, y para el diente de sierra basta con tiempos diferentes.

señal perfectamente cuadrada no es posible con este generador. En el caso de que necesitemos lograr una señal de este tipo, deberemos obtenerla con algún generador digital.

GUÍA VISUAL ■ CONFIGURACIÓN DE PULSE



01 **INITIAL (LOW) VOLTAGE (VOLTAJE INICIAL BAJO):** este es el voltaje bajo inicial del pulso, marcado en la **Figura 8** como **V1**.

02 **PULSED (HIGH) VOLTAGE (VOLTAJE DE PULSO ALTO):** en este campo definimos el voltaje alto del pulso marcado en la **Figura 8** como **V2**.

03 **START (SECS) (TIEMPO DE INICIO):** aquí especificamos un tiempo de inicio, es decir, lo que tardará en darse el primer pulso, en segundos.

04 **RISE TIME (SECS) (TIEMPO DE SUBIDA):** es el tiempo que tarda la señal en pasar del voltaje bajo al alto.

05 **FALL TIME (SECS) (TIEMPO DE BAJADA):** es el tiempo que tarda la señal en pasar del voltaje alto al bajo al terminar el pulso.

06 **PULSE WIDTH (SECS) (ANCHO DE PULSO):** para especificar el ancho del pulso en el estado alto, en segundos.

07 **PULSE WIDTH (%) (ANCHO DE PULSO (%)):** aquí es posible indicar el ancho del pulso en el estado alto, en porcentaje con respecto al período de la señal.

08 **FREQUENCY (HZ) (FRECUENCIA):** este campo permite definir la frecuencia de la señal, ya que se generará un tren de pulsos continuo al elegir esta opción.

09 **PERIOD (SECS) (PERÍODO):** aquí establecemos el período en segundos; en la **Figura 8** este parámetro está marcado como **P**.

10 **CYCLES/GRAPH (CICLOS POR GRÁFICO):** este campo permite generar un número determinado de pulsos en cierto tiempo, esto depende de un gráfico de simulación. Hablaremos sobre este tema en el **Capítulo 5**.

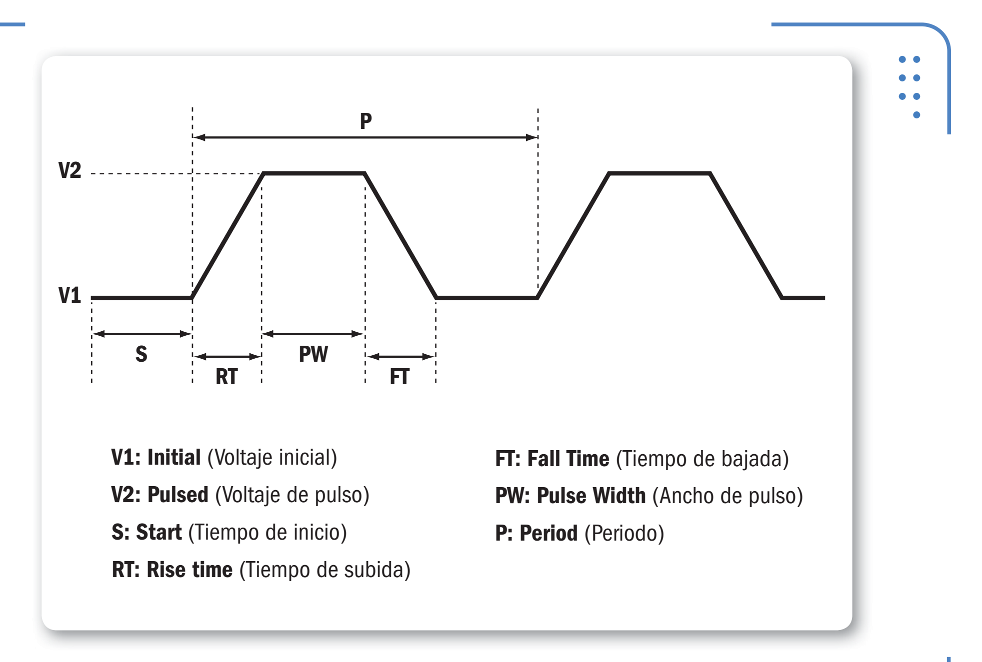


Figura 8. Podemos construir un pulso analógico definiendo sus parámetros con el generador PULSE.

Generador exponencial (EXP)

El generador **EXP** es capaz de formar una señal exponencial imitando la carga y descarga en un **circuito RC**.

V2
0.63V2
0.37V2
V1
RS
TAU1
TAU2
FS

V1: Initial (Voltaje inicial)
V2: Pulsed (Voltaje de pulso)
RS: Rise Start (Tiempo de inicio de subida)
FS: Fall Start (Tiempo de incio de bajada)
TAU1: Constante de tiempo para la curva de subida
TAU2: Constante de tiempo para la curva de bajada

Figura 9. Podemos construir una señal exponencial definiendo sus parámetros con el generador EXP.

En la **Figura 9** vemos una ilustración de este tipo de señal y los parámetros que debemos considerar para construirla con un generador EXP. A continuación, en la siguiente **Guía visual**, explicaremos en detalle las diferentes opciones de configuración.

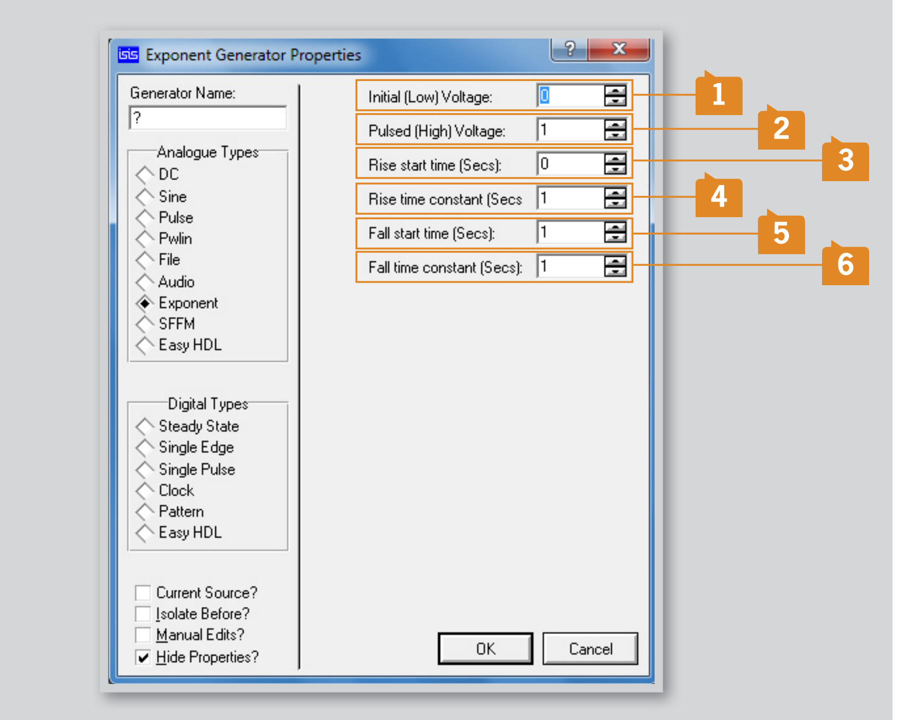


¿BATERÍAS, TERMINALES O GENERADORES?

Existen múltiples maneras de generar voltajes de corriente directa. Es recomendable utilizar baterías cuando el circuito realmente será alimentado con ellas. Por lo demás, es lo mismo usar un método u otro. Para conocar más opciones, tenemos una fuente de voltaje y una de corriente en la categoría **Simulator Primitives**, son **VSOURCE** y **CSOURCE**.

▼ GUÍA VISUAL ■ CONFIGURACIÓN DE EXP





01 **INITIAL (LOW) VOLTAGE (VOLTAJE INICIAL EN BAJO):** es el voltaje bajo inicial del pulso, marcado en la **Figura 9** como **V1**.

02 **PULSED (HIGH) VOLTAGE (VOLTAJE DE PULSO EN ALTO):** en este campo definimos el voltaje alto del pulso, marcado en la **Figura 9** como **V2**.

03 **RISE START TIME (SECS) (TIEMPO DE INICIO DE SUBIDA):** tiempo en el que se iniciará la subida de la curva.

04 **RISE TIME CONSTANT (SECS) (CONSTANTE DE TIEMPO DE SUBIDA):** tiempo que tarda la señal en subir hasta aproximadamente el 63% del voltaje más alto (**TAU1**).

05 **FALL START TIME (SECS) (TIEMPO DE INICIO DE BAJADA):** tiempo en que comienza la curva de bajada de la señal; se mide desde el inicio.

06 **FALL TIME CONSTANT (SECS) (CONSTANTE DE TIEMPO DE BAJADA):** es el tiempo que tarda la señal en bajar hasta un voltaje aproximado del 37% del voltaje más alto (**TAU2**).

Generador de frecuencia modulada (SFFM)

El generador **SFFM** produce una forma de onda que es el resultado de una señal senoidal modulada en frecuencia con otra señal senoidal, dados los parámetros de las ondas y el índice de modulación. En la siguiente **Guía visual** explicaremos las opciones de configuración.

▼ GUÍA VISUAL ■ CONFIGURACIÓN DE SFFM

01 **OFFSET (VOLTS) (VOLTAJE OFFSET):** define el voltaje offset, también conocido como voltaje de desplazamiento vertical.

02 **AMPLITUDE (VOLTS) (AMPLITUD):** determina la amplitud o voltaje pico de la onda resultante.

03 **CARRIER FREQ. (HZ) (FRECUENCIA DE LA PORTADORA):** establece la frecuencia en Hertz de la señal senoidal que servirá de portadora.



MODULATION INDEX (ÍNDICE DE MODULACIÓN): permite indicar el índice de modulación.



SIGNAL FREQ. (HZ) (FRECUENCIA DE LA SEÑAL): aquí definiremos la frecuencia de la señal moduladora en Hertz.

Generador de señales lineales complejas (PWLIN)

PWLIN es un generador de señales complejas lineales, que puede ser usado cuando no es posible producir una señal personalizada con el generador de pulsos. El nombre significa **Piece-wise linear**, que puede traducirse como **lineal por intervalos o piezas**. A continuación, conoceremos sus opciones de configuración.

▼ GUÍA VISUAL ■ CONFIGURACIÓN DEL PWLIN

01 **EXPAND GRAPH (EXPANDIR GRÁFICO):** permite expandir el gráfico para trabajar más cómodamente.

02 **TIME/VOLTAGES (TIEMPO/VOLTAJES):** mediante este gráfico, es posible construir la señal colocando puntos con clics del mouse.

03 **SCALING (ESCALAS):** desde aquí podemos cambiar las escalas del eje X o del eje Y en el gráfico para ampliarlas o reducirlas; el eje **X** representa el tiempo, y el **Y**, la amplitud.

04 **MINIMUM RISE/FALL TIME (SECS) (TIEMPO MÍNIMO DE SUBIDA O BAJADA):** en este campo es posible establecer el tiempo mínimo para un flanco de subida o bajada. Podemos cambiar su valor y, al intentar colocar un punto justo arriba o debajo de otro en el gráfico, el tiempo de subida o bajada se ajustará con ese parámetro.

05 **USE (USO):** aquí se dan las instrucciones de uso del gráfico.

- Clic izquierdo del mouse para colocar/arrastrar un punto.
- Clic derecho del mouse para borrar un punto.
- **CTRL** + clic derecho para borrar todos los puntos. Al usar esta última opción, se pedirá confirmación para eliminar todos los puntos del gráfico.

Podemos construir una señal definiendo puntos en el gráfico; cada punto se unirá con el anterior y el siguiente mediante líneas rectas. **Expand Graph** abre una nueva ventana con el gráfico ampliado, que puede ser redimensionada o maximizada para verla en pantalla completa.

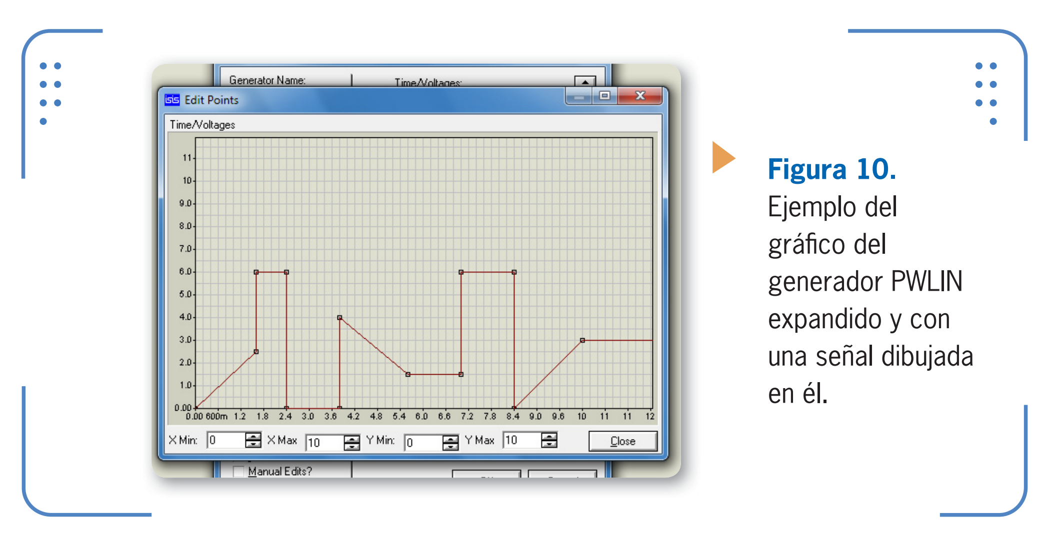
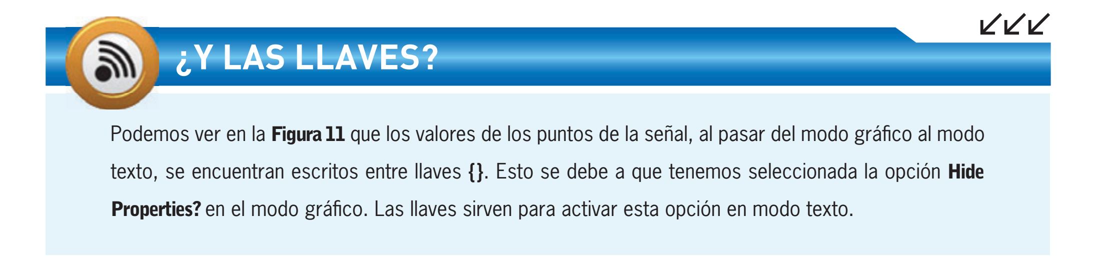


Figura 10. Ejemplo del gráfico del generador PWLIN expandido y con una señal dibujada en él.

En **Edit Points**, vemos en la parte inferior los controles para cambiar la escala de los ejes. Si marcamos **Manual Edits?** en las propiedades del generador, podemos editar los puntos del gráfico de modo manual con el formato para cada punto: **V(t)=A**, donde **t** es el tiempo de un punto, y **A**, la amplitud de ese mismo punto. Por ejemplo: **V(2)=3** representa un punto colocado a los **2 segundos** con una amplitud de **3 Volts**.

Figura 11. Ejemplo del mismo gráfico de la **Figura 10** expresado como texto.

Es posible alternar entre el modo de texto y el modo gráfico en cualquier momento para editar la señal en ambos modos.



¿Y LAS LLAVES?

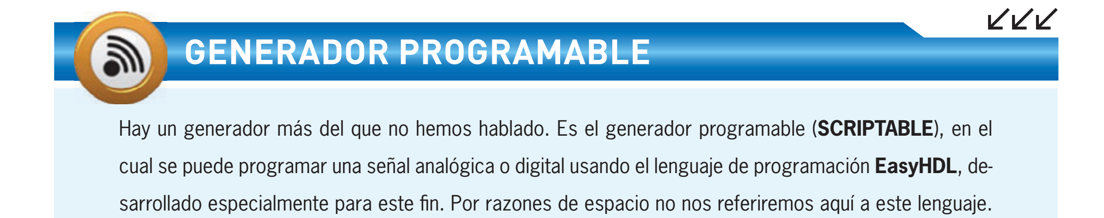
Podemos ver en la **Figura 11** que los valores de los puntos de la señal, al pasar del modo gráfico al modo texto, se encuentran escritos entre llaves **{}**. Esto se debe a que tenemos seleccionada la opción **Hide Properties?** en el modo gráfico. Las llaves sirven para activar esta opción en modo texto.

Generador con archivos (FILE)

Una variante para producir señales punto a punto o por intervalos es con el generador **FILE**, que utiliza datos extraídos de un archivo en formato ASCII para construir una señal.

Figura 12. En las propiedades del generador FILE definiremos el archivo con los datos de la señal.

Si la señal tiene muchos puntos, tal vez resulte mejor usar un generador FILE en vez de PWLIN, además de que el archivo puede guardarse para usar en otras simulaciones. Dentro de las propiedades del generador FILE, encontraremos el campo **Data File** y el botón **Browse**;



GENERADOR PROGRAMABLE

Hay un generador más del que no hemos hablado. Es el generador programable (**SCRIPTABLE**), en el cual se puede programar una señal analógica o digital usando el lenguaje de programación **EasyHDL**, desarrollado especialmente para este fin. Por razones de espacio no nos referiremos aquí a este lenguaje.

al presionarlo, accederemos al explorador para encontrar el archivo donde están los datos de la señal que queremos generar.

El formato para generar un archivo de señal es el siguiente: cada punto se define por un par de valores; el primero determina el **tiempo**, y el segundo, la **amplitud**. Estos valores deben estar separados por espacios o tabuladores, y cada par tiene que estar en una línea; es decir, cada par se separa del anterior y del siguiente mediante saltos de línea. Por ejemplo, los siguientes valores generarán una señal triangular de 5 ciclos, un período de 20 ms, con picos de 3 Volts:

```
0       0
10e-3   3
20e-3   0
30e-3   3
40e-3   0
50e-3   3
60e-3   0
70e-3   3
80e-3   0
90e-3   3
10e-2   0
```

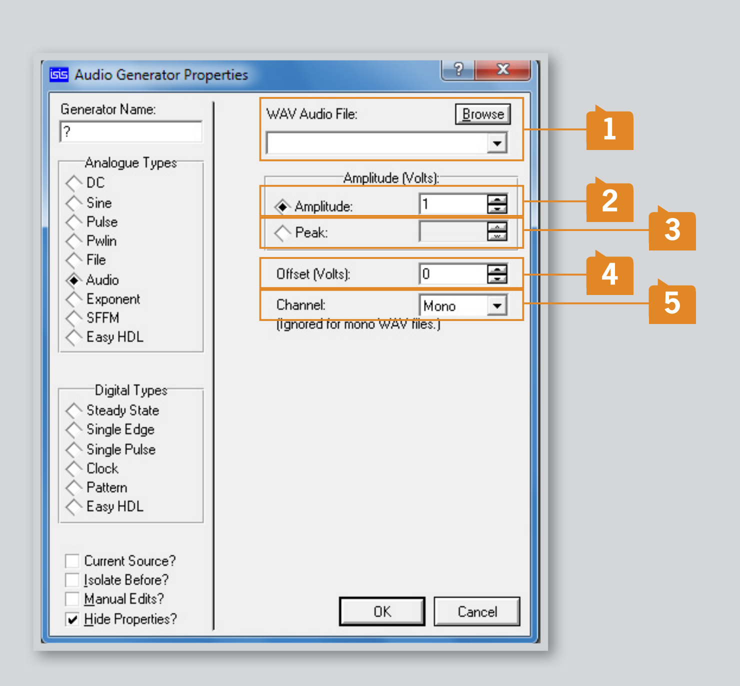
En este ejemplo, el primer valor (**0 0**) define el primer punto en el tiempo 0 y una amplitud de 0 Volts. El segundo (**10e-3 3**) determina el segundo punto en un tiempo de 10 milisegundos con una amplitud de 3 Volts. Como podemos observar, es posible usar un formato de coma flotante para los valores. Podemos escribir los valores anteriores en un documento del Bloc de notas siguiendo los lineamientos y, depués, guardarlo como un archivo de texto (***.TXT**), y ya tendremos la señal.

Generador de audio (AUDIO)

En algunas de las simulaciones necesitaremos usar señales de audio para introducir en los circuitos, por ejemplo en un amplificador, para lo cual podemos auxiliarnos del generador de **audio**. Debemos tener un archivo de audio en formato **.WAV** para poder utilizar este generador. En la **Guía visual** explicaremos las opciones de configuración.

GUÍA VISUAL ■ CONFIGURACIÓN DE AUDIO





01 **WAV AUDIO FILE (ARCHIVO DE AUDIO WAV):** permite elegir el archivo de audio. Al presionar **Browse**, podemos navegar en nuestro sistema hasta encontrar el archivo en formato **WAV** deseado.

02 **AMPLITUDE (AMPLITUD):** en este campo podemos definir la amplitud que tendrá la señal de audio, en Volts.

03 **PEAK (PICO A PICO):** para configurar la amplitud de la señal en Volts pico a pico.

04 **OFFSET (VOLTS) (VOLTAJE OFFSET):** al igual que en otros generadores, este valor define el voltaje offset, también conocido como voltaje de desplazamiento vertical de la señal.

05 **CHANNEL (CANAL):** permite elegir el canal de audio que queremos utilizar. Si el archivo **.WAV** elegido es **monoaural**, no importa la selección de esta lista, porque se usará el único canal disponible. Si el archivo **.WAV** es **estéreo**, se puede elegir el canal izquierdo (**Left**) o el derecho (**Right**).

Proyecto: vúmetro con LM3915

Veamos un sencillo ejemplo del uso del generador de audio. Esta vez simularemos un **vúmetro** usando el circuito integrado **LM3915**.

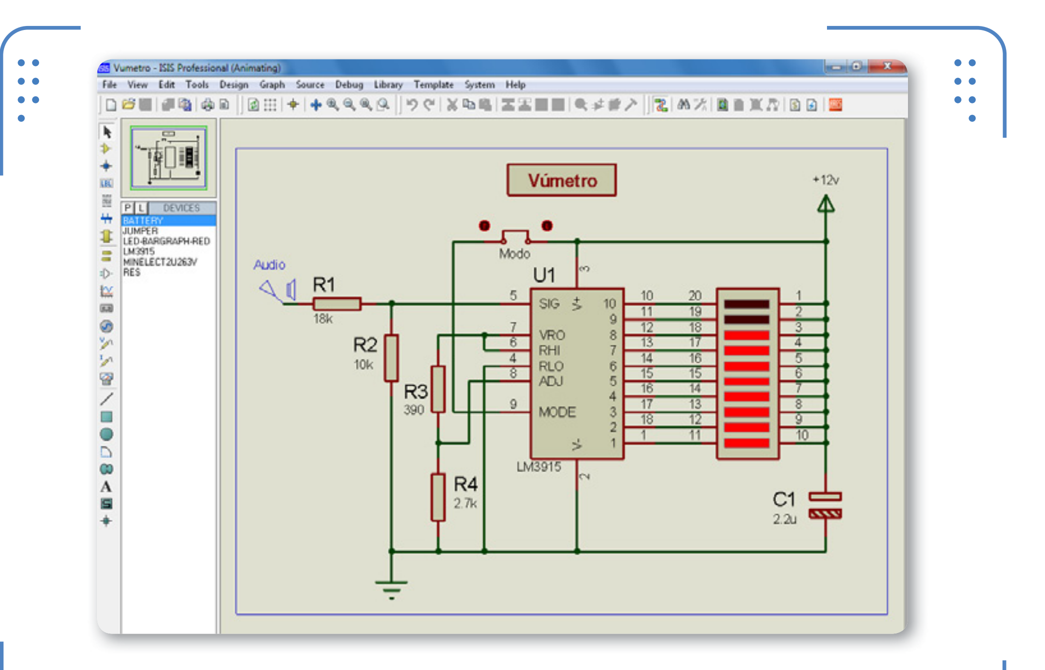


Figura 13. Es posible simular el funcionamiento de un vúmetro con la ayuda del generador de audio.

Para observar esta simulación, descargamos los archivos **Vumetro.dsn** y **Audio01.wav**, que contiene un pequeño fragmento de música. Hemos usado un jumper activo para cambiar de modo, ya sea punto o barra. Como el generador de audio requiere muchos recursos de procesamiento, la simulación seguramente no correrá en tiempo



WAV EN LA MISMA CARPETA

Si vamos a la ventana de propiedades del generador de audio en la simulación del vúmetro, veremos que el archivo de audio no tiene ruta, solo se muestra su nombre. Esto es porque podemos colocarlo en la misma carpeta donde está el **.DSN** y, de esta manera, se lo localizará con solo colocar el nombre.

real debido a la alta **carga de la CPU**. Este generador puede ser de utilidad para analizar los circuitos en forma gráfica en vez de intentar simularlos en tiempo real; en el **Capítulo 5** desarrollaremos este tema.

Generador de estados digitales (DSTATE)

DSTATE es el primer generador digital que analizaremos. Como su nombre lo indica, sirve para producir estados lógicos en los circuitos digitales. Aunque podríamos pensar que este generador tiene solo dos estados (alto y bajo), en realidad cuenta con siete estados diferentes, que explicaremos a continuación.

GENERADOR DSTATE

ESTADO	DESCRIPCIÓN
Power Rail High o PHI (alto a línea de alimentación)	Estado alto conectado directamente a una fuente de alimentación (VCC o VDD).
Strong High o SHI (alto fuerte)	Estado alto equivalente a salida activa.
Weak High o WHI (alto débil)	Estado alto equivalente a salida pasiva.
Floating o FLT (flotante)	Salida flotante o alta impedancia.
Weak Low o WLO (bajo débil)	Estado bajo equivalente a salida pasiva.
Strong Low o SLO (bajo fuerte)	Estado bajo equivalente a salida activa.
Power Rail Low o PLO (bajo a línea de alimentación)	Estado bajo conectado directamente a fuente de alimentación (tierra).

Tabla 1. Estados del generador DSTATE.

Generador de flancos (DEDGE)

El generador digital **DEDGE** sirve para producir un cambio de estado o flanco, ya sea de bajada o de subida. A continuación, en la **Guía visual**, veremos las opciones de configuración.

GUÍA VISUAL ■ CONFIGURACIÓN DE DEDGE

01 **POSITIVE (LOW-TO-HIGH) EDGE (FLANCO DE SUBIDA):** en la sección **Edge Polarity**, al marcar esta opción, elegiremos la generación de un flanco de subida.

02 **NEGATIVE (HIGH-TO-LOW) EDGE (FLANCO DE BAJADA):** con esta opción elegiremos un flanco de bajada.

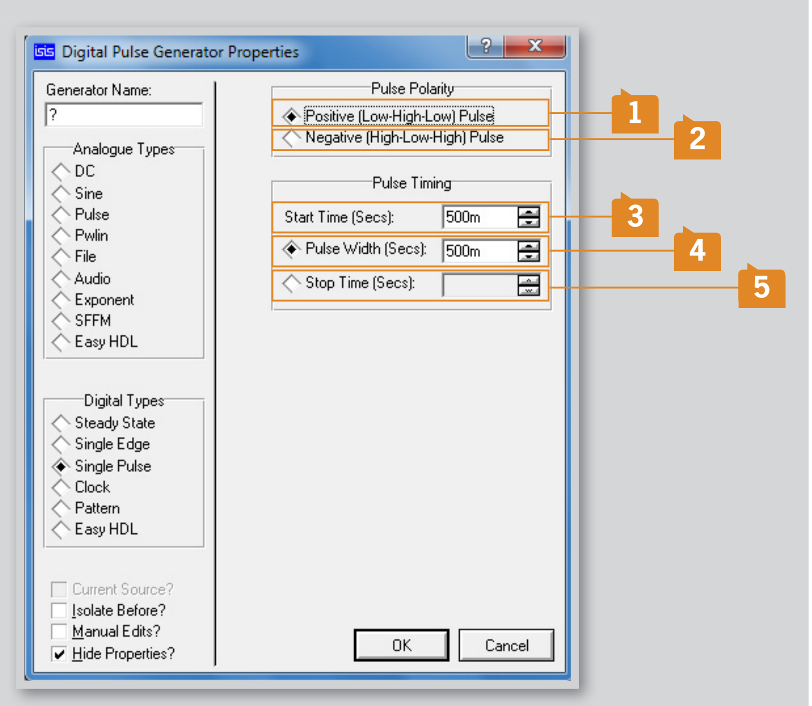
03 **EDGE AT (SECS) (TIEMPO DEL FLANCO):** en este campo debemos especificar el tiempo en el cual el flanco ocurrirá.

Generador de pulso simple (DPULSE)

El generador de pulso simple, que encontramos en la lista bajo el nombre de **DPULSE**, puede dar un **pulso digital** simple, es decir, de un solo pulso. En la siguiente **Guía visual** vamos a explicar cómo debemos proceder para configurarlo.

GUÍA VISUAL ■ CONFIGURACIÓN DE DPULSE





01 **POSITIVE (LOW-HIGH-LOW) PULSE (PULSO POSITIVO - BAJO-ALTO-BAJO):** producirá un pulso positivo que comienza en bajo, va al estado alto por un tiempo determinado y regresa al estado bajo.

02 **NEGATIVE (HIGH-LOW-HIGH) PULSE (PULSO NEGATIVO - ALTO-BAJO-ALTO):** generará un pulso negativo contrario al anterior.

03 **START TIME (SECS) (TIEMPO DE INICIO):** aquí podemos colocar el tiempo en segundos en que el pulso comenzará.

04 **PULSE WIDTH (SECS) (ANCHO DE PULSO):** permite definir el tiempo que durará el pulso, en segundos.

05 **STOP TIME (SECS) (TIEMPO DE FINALIZACIÓN):** mediante esta opción, estableceremos el tiempo en el que el pulso terminará. A diferencia de la opción anterior, este se cuenta desde cero.

Generador de reloj (DCLOCK)

El generador **DCLOCK** entregará una señal de **reloj digital**, es decir, un tren de pulsos cuadrados continuo. En la siguiente **Guía visual** conoceremos las opciones de configuración.

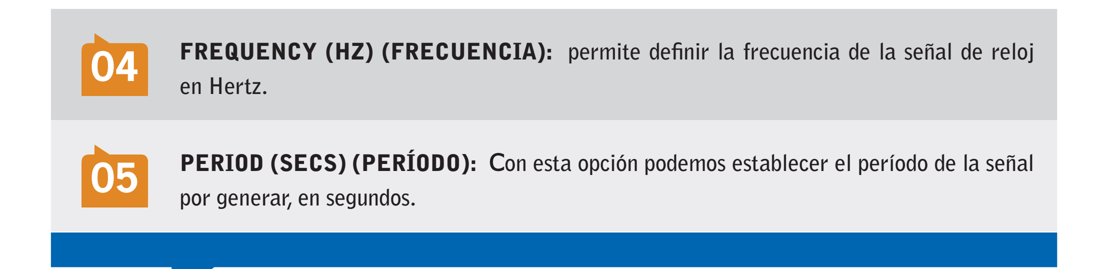
▼ GUÍA VISUAL ■ CONFIGURACIÓN DE DCLOCK



01 **LOW-HIGH-LOW CLOCK (RELOJ BAJO-ALTO-BAJO):** con esta opción se genera una señal que va de bajo a alto, y luego, a bajo.

02 **HIGH-LOW-HIGH CLOCK (RELOJ ALTO-BAJO-ALTO):** permite producir una señal que va de alto a bajo, y luego, a alto.

03 **FIRST EDGE AT (TIEMPO DEL PRIMER FLANCO):** aquí podemos ingresar el tiempo en segundos en que la señal de reloj comenzará.



04 **FREQUENCY (HZ) (FRECUENCIA):** permite definir la frecuencia de la señal de reloj en Hertz.

05 **PERIOD (SECS) (PERÍODO):** Con esta opción podemos establecer el período de la señal por generar, en segundos.

Debemos saber que la señal generada tendrá un ciclo activo del 50%. Con esto podría parecer inútil elegir entre las opciones **Low-High-Low Clock** y **High-Low-High Clock**, pero lo que esto determinará es el nivel de la señal al inicio, sobre todo, en combinación con la opción **First Edge At** al colocar un valor mayor que 0 en ella.

Generador de patrones digitales (DPATTERN)

El generador **DPATTERN** permite construir una señal digital arbitraria o personalizada. Puede usarse para formar, por ejemplo, un tren de pulsos con un determinado número de pulsos y, además, con temporización diferente para cada uno. A continuación, veremos las opciones de configuración.

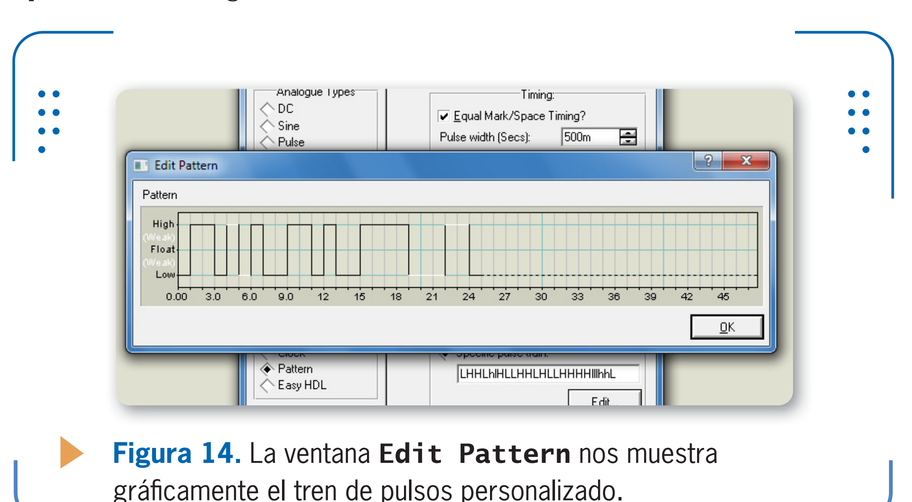
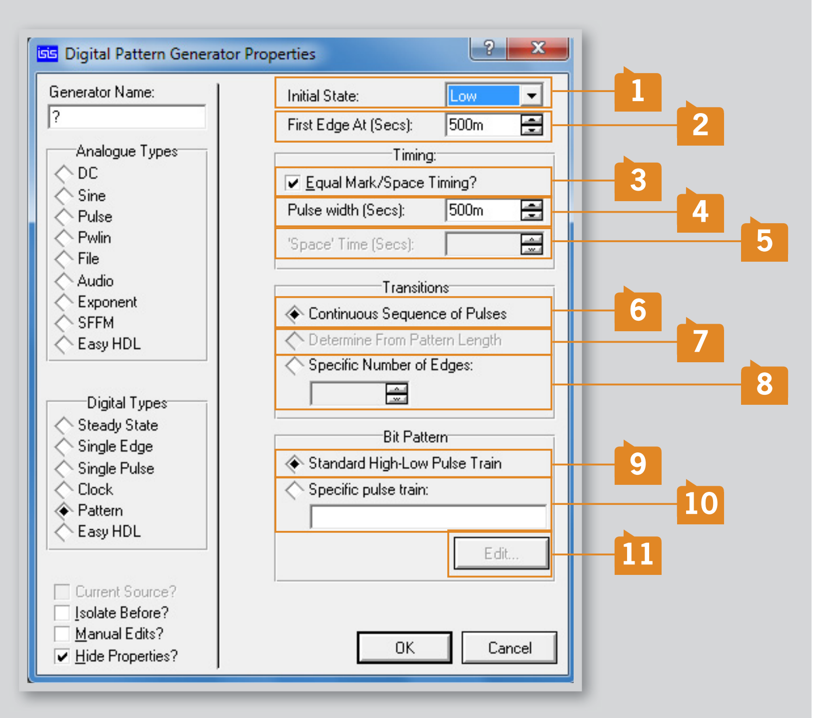


Figura 14. La ventana **Edit Pattern** nos muestra gráficamente el tren de pulsos personalizado.

▼ GUÍA VISUAL ■ CONFIGURACIÓN DEL DPATTERN





01 **INITIAL STATE (ESTADO INICIAL):** en esta lista podemos elegir el estado inicial de la señal.

02 **FIRST EDGE AT (SECS) (TIEMPO DEL PRIMER FLANCO):** aquí es posible definir el tiempo en que ocurrirá el primer flanco.

03 **EQUAL MARK/SPACE TIMING? (¿IGUAL TIEMPO EN ALTO/BAJO?):** permite especificar si la señal tendrá tiempos iguales en los estados altos y bajos. Si la marcamos, solo podremos editar uno de los tiempos, que será el mismo para ambos estados.

04 **PULSE WIDTH (SECS):** aquí definimos la duración de los pulsos en el estado alto, en segundos.

05 **'SPACE' TIME (SECS) (TIEMPO EN BAJO):** aquí determinamos la duración de los pulsos en el estado bajo, en segundos. Si activamos **Equal Mark/Space Timing?**, esta opción estará deshabilitada.

06 **CONTINUOUS SEQUENCE OF PULSES (SECUENCIA DE PULSOS CONTINUA):** al seleccionar esta opción, se genera un tren de pulsos continuo, es decir, sin fin.

07 **DETERMINE FROM PATTERN LENGTH (DETERMINADO POR EL PATRÓN):** permite generar un número de pulsos determinado por la opción **Specific Pulse Train**.

08 **SPECIFIC NUMBER OF EDGES (NÚMERO ESPECÍFICO DE FLANCOS):** aquí definimos cuántos flancos tendrá la señal.

09 **STANDARD HIGH-LOW PULSE TRAIN (TREN DE PULSOS ESTÁNDAR):** si activamos esta opción, se genera un tren de pulsos estándar.

10 **SPECIFIC PULSE TRAIN (TREN DE PULSOS ESPECÍFICO):** cuando marcamos esta opción, podemos definir un tren de pulsos personalizado.

11 **EDIT (EDITAR):** al presionar este botón, se abre una ventana donde es posible editar el tren de pulsos de manera gráfica.

Si marcamos la opción **Specific Pulse Train**, podemos ingresar una secuencia de pulsos en el **campo de texto** de la siguiente forma: H=alto, h=alto débil, f o F=flotante, l=bajo débil, L=bajo. Los tiempos de cada estado estarán determinados por los valores que hayamos elegido en la sección **Timing**. Si presionamos **Edit**, se abrirá la ventana **Edit Pattern**, en la cual podemos editar de forma gráfica los pulsos.

En el gráfico, la cuadrícula también está determinada por los valores de la sección **Timing**. Podemos editar el patrón desde el cuadro de texto de la opción **Specific Pulse Train** o desde el gráfico. En el gráfico los estados débiles se muestran con líneas blancas.

RESUMEN

Así como en la realidad existen diferentes formas de obtener señales para los circuitos electrónicos, desde un voltaje de corriente directa hasta señales complejas, en Proteus también encontramos múltiples posibilidades. En este capítulo estudiamos cómo se configuran las líneas o fuentes de alimentación de corriente directa en ISIS para la alimentación de los circuitos. Además, aprendimos a usar los diferentes generadores de señales con el fin de contar con una gran variedad de señales para las simulaciones.

Actividades

TEST DE AUTOEVALUACIÓN

1 ¿Cómo podemos ver el nombre de los pines ocultos de un circuito integrado?

2 ¿Cuántas son las líneas de alimentación por defecto y cómo se llaman?

3 ¿Cuál es el voltaje por defecto de la línea VCC/VDD?

4 ¿Cómo se debe nombrar una terminal POWER para obtener un voltaje de 24 Volts?

5 ¿Para qué sirven los generadores?

6 ¿Cómo se entra en modo G-enerador?

7 Mencione dos generadores analógicos.

8 Mencione dos generadores digitales.

9 ¿Cómo se puede convertir un generador analógico en una fuente de corriente?

10 ¿Con qué generador es posible obtener una señal de reloj para los circuitos digitales?

ACTIVIDADES PRÁCTICAS

1 Dibuje el circuito de la Figura 2 y compruebe su funcionamiento.

2 Dibuje el circuito de la Figura 3 y compruebe su funcionamiento.

3 Abra el archivo **Vumetro.dns** y reemplace el generador de audio por un PULSE con los siguientes parámetros: Initial=0, Pulse=30, Start=0, Rise Time=500m, Fall Time=500m, Pulse Width (Secs)=0, Period=1.

4 Compruebe la simulación y note cómo la carga de la CPU es mucho menor que con el generador de audio.

5 Simule de nuevo el vúmetro usando un archivo **WAV** diferente del que se proporciona como ejemplo.



Instrumentos de medición virtuales

Así como en la realidad disponemos de múltiples instrumentos de medición en nuestro laboratorio de electrónica, en el módulo ISIS también contamos con gran variedad de herramientas virtuales de medición y análisis. En este capítulo estudiaremos qué instrumentos de medición podemos usar en Proteus. Su funcionamiento y manejo son similares a los reales, por lo que será fácil aprender a usarlos.



↗↗↗



Los instrumentos de medición virtuales

ISIS cuenta con múltiples **instrumentos virtuales** de medición y análisis, que imitan a las herramientas que usamos en nuestro laboratorio día a día. La forma de utilizarlos y sus funciones son muy similares a las de los instrumentos reales, lo cual nos permite tener todo un laboratorio virtual en nuestra computadora.

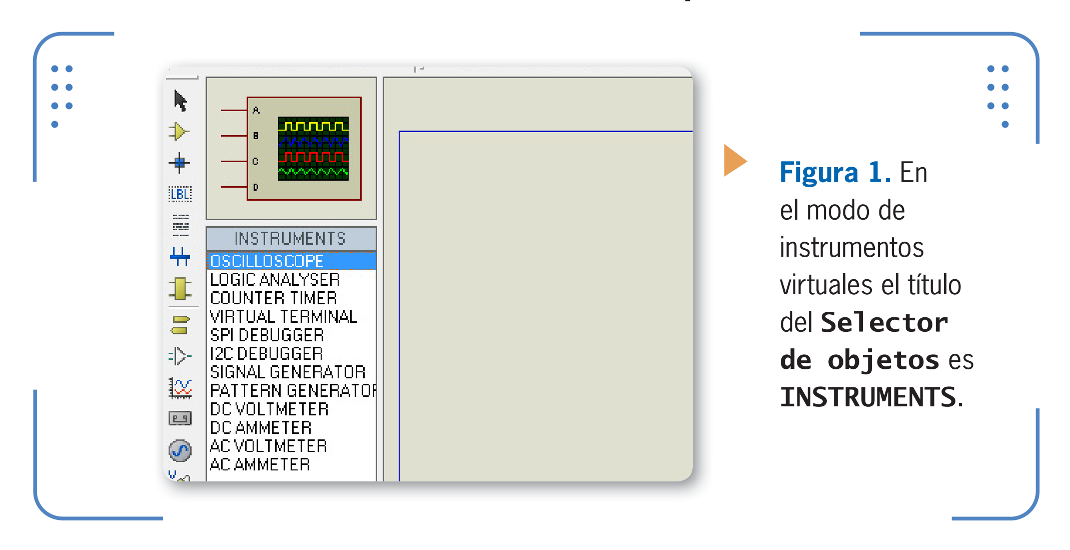


Figura 1. En el modo de instrumentos virtuales el título del **Selector de objetos** es **INSTRUMENTS**.

Para acceder a los instrumentos virtuales en ISIS, tenemos un botón en la barra de herramientas de **Modo** con la forma de un pequeño medidor analógico, llamado **Virtual Instruments Mode**. Al presionarlo, entraremos en el modo de instrumentos virtuales, y el **Selector de objetos** mostrará la lista con los instrumentos disponibles.



VIRTUALES VS. REALES

La intención de los instrumentos virtuales es que sean muy parecidos a los reales, por lo que su uso es muy similar. Así que los voltímetros y amperímetros deben colocarse de la misma forma como se hace en la vida real: los voltímetros en paralelo, y los amperímetros en serie en los puntos de medición.





Voltímetros y amperímetros

Algunos de los instrumentos que utilizamos con mayor frecuencia, al trabajar con circuitos, suelen ser los voltímetros y amperímetros. Estos se encuentran al final de la lista y son: voltímetro de corriente directa (**DC VOLTMETER**), voltímetro de corriente alterna (**AC VOLTMETER**), amperímetro de corriente directa (**DC AMMETER**) y amperímetro de corriente alterna (**AC AMMETER**).

Si vamos a las propiedades del voltímetro de corriente directa, veremos dos campos: **Display Range** para elegir el rango, que puede ser **volts**, **milivolts** o **microvolts**; y **Load Resistance** para seleccionar la resistencia interna del voltímetro, que por defecto es **100M**. En el voltímetro de corriente alterna, además, disponemos de la opción **Time Constant**, para definir la constante de tiempo. En los amperímetros también tenemos la posibilidad de cambiar el rango en **amperes**, **miliamperes** o **microamperes**.

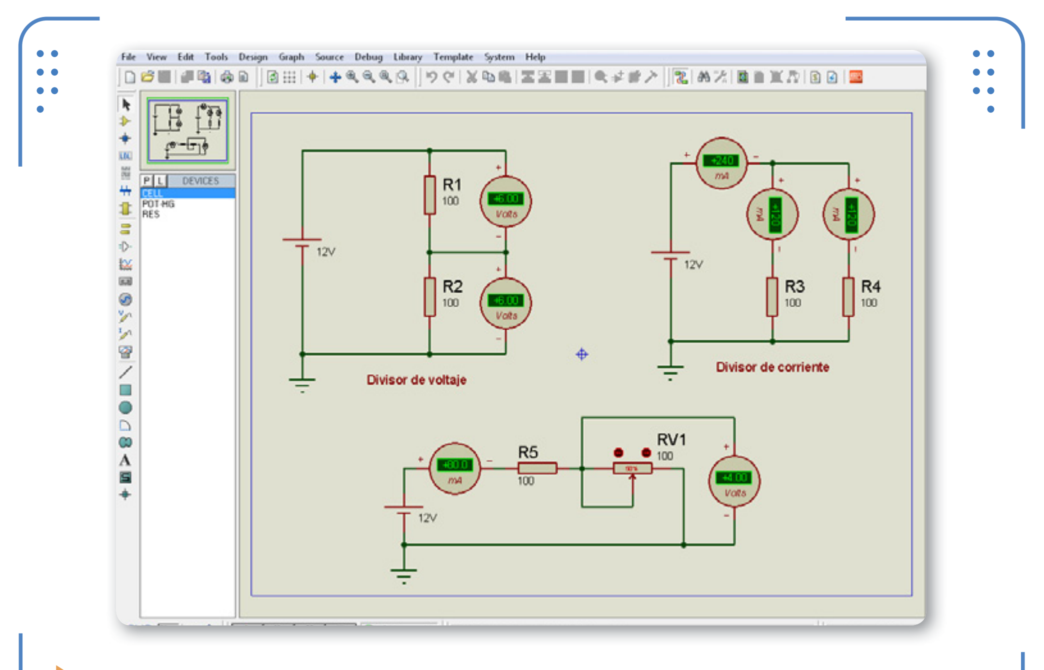


Figura 2. Algunos ejemplos de divisores de voltaje y corriente con instrumentos virtuales en ISIS.

Tomemos como ejemplo el archivo **Divisores.dsn**, que estudiamos en el **Capítulo 2**, donde usamos sondas de voltaje y corriente. Esta vez cambiaremos las sondas por voltímetros y amperímetros. Podemos ver esta nueva simulación en el archivo **Divisores2.dsn**.

Debemos saber que, tal como ocurre en la realidad, el voltímetro de corriente alterna de ISIS mide los valores en **RMS**.



Osciloscopio virtual

El módulo ISIS cuenta con un **osciloscopio digital virtual** de cuatro canales que permite ver las formas de onda generadas en los circuitos que simulemos.

Para usarlo, basta con seleccionar **OSCILLOSCOPE** en la lista de instrumentos virtuales y colocarlo en el diseño como si fuera un componente más. En la **Figura 3** vemos un ejemplo.

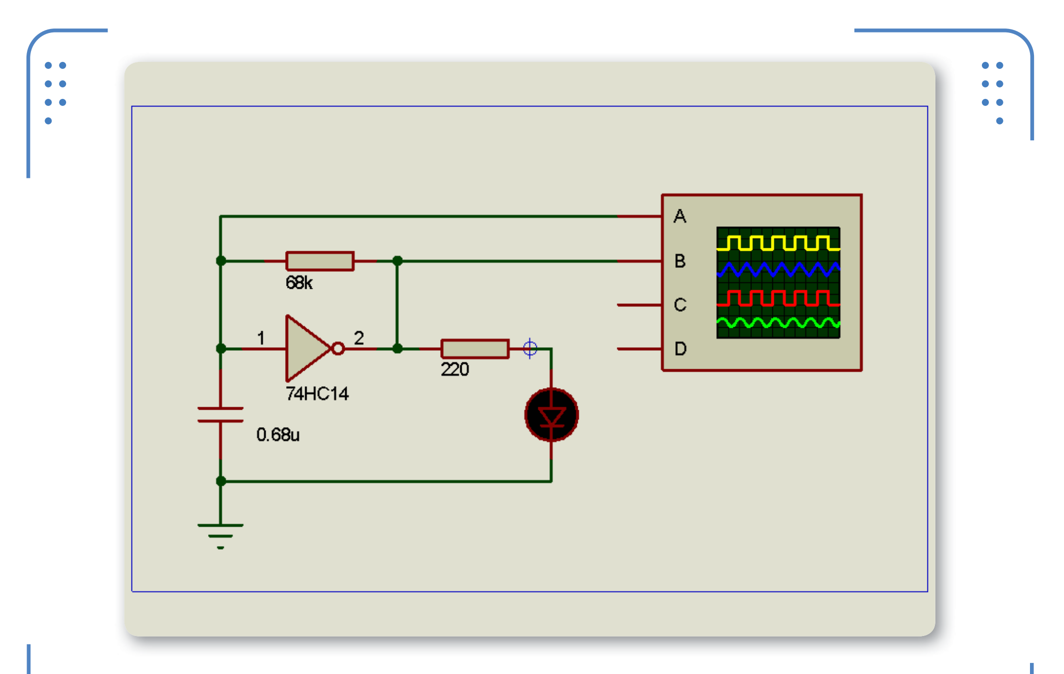
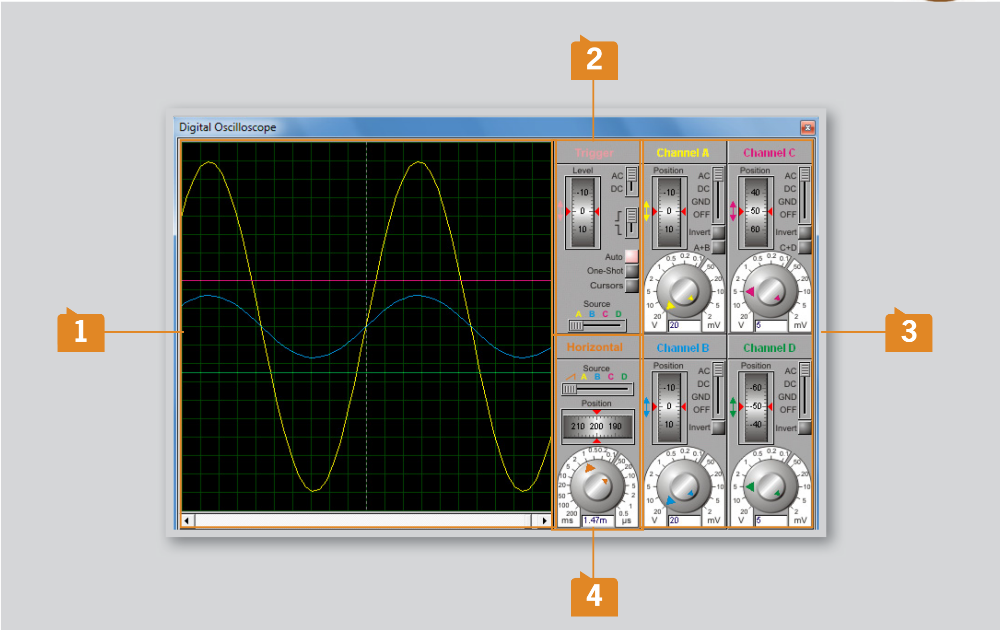


Figura 3. El osciloscopio virtual en ISIS se coloca y conecta como cualquier otro componente.

El símbolo del osciloscopio contiene las terminales de conexión de los cuatro canales, **A**, **B**, **C** y **D**, que se conectan a los puntos donde tomaremos la señal que vamos a medir. Al correr la simulación después de haber agregado y conectado el osciloscopio, aparecerá el instrumento completo en una pantalla más grande con el título **Digital Oscilloscope**. En la siguiente **Guía visual** conoceremos cada uno de sus elementos.

▼ GUÍA VISUAL ■ CONFIGURACIÓN DEL OSCILOSCOPIO





01 **PANTALLA:** en ella vemos las señales que vamos a medir; cada canal tiene un color diferente.

02 **TRIGGER (DISPARO):** esta es la sección de disparo, muy similar a la de un osciloscopio real.

03 **CANALES:** incluye los controles para los cuatro canales del osciloscopio. En cada uno podemos controlar la posición y la escala vertical.

04 **HORIZONTAL (AJUSTE HORIZONTAL):** permite controlar la posición y la escala horizontal en la pantalla.

Las funciones del osciloscopio virtual son idénticas a las de uno real, así que será muy fácil acostumbrarse a usar este instrumento en Proteus.

Cursores

En la sección **Trigger** del osciloscopio, hay una opción llamada **Cursors**, que permite utilizar cursores en la pantalla para realizar mediciones. Si la activamos, notaremos que al colocar el cursor del mouse en la pantalla del osciloscopio, mientras está corriendo una simulación, aparecen dos líneas, una en sentido vertical y otra horizontal, que lo siguen en todo momento e indican la posición horizontal (tiempo) y el voltaje de las señales.

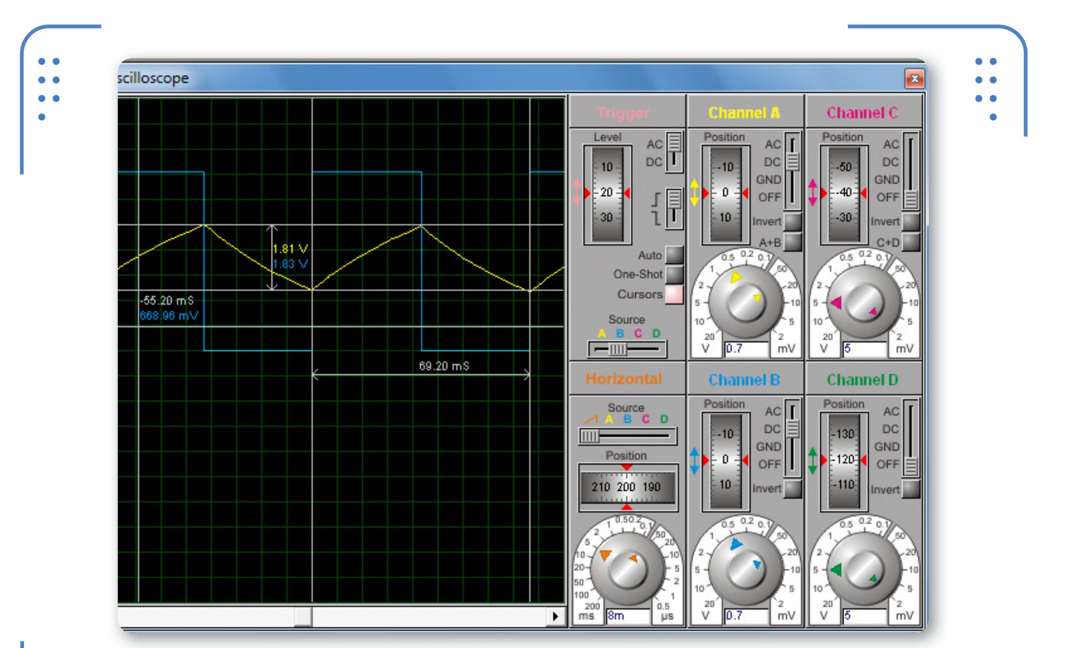


Figura 4. Los cursores permiten obtener medidas de tiempo y voltajes directamente en la pantalla del osciloscopio.

Si hacemos un clic en algún lugar de la pantalla y arrastramos hacia arriba o hacia abajo, aparece una flecha vertical que mide el voltaje comprendido en ese intervalo. Esto puede ser útil para conocer los voltajes de las señales en pantalla. Si arrastramos horizontalmente,

se mide el intervalo de tiempo; así obtendremos, por ejemplo, los períodos de las señales.

Podemos mover las flechas o los cursores simplemente colocándonos sobre ellos y arrastrándolos para modificar su posición en cualquier momento. Si hacemos un clic con el botón derecho del mouse sobre un cursor, se abre un menú contextual que nos permitirá borrarlo (**Delete Cursor**) y también eliminar todos los cursores en la pantalla (**Clear All Cursors**). Para ocultar o mostrar los cursores en cualquier momento, solo debemos presionar al botón **Cursors**.

Imprimir y personalizar la pantalla del osciloscopio

Al hacer un clic con el botón derecho del mouse en cualquier parte del osciloscopio, se abre un menú contextual donde encontramos la opción **Setup**, que permite personalizar los colores en la pantalla. Al presionarla, aparece una ventana para elegir los colores del fondo, de los cuatro canales y de los cursores. Podemos cambiar tanto los colores que se mostrarán en el monitor de nuestra computadora (**Display**) como los que saldrán en una impresión (**Printer**). Si seleccionamos la opción **Black and White**, las impresiones saldrán en blanco y negro. Por último, el botón **Reset** devolverá los colores por defecto.

Para imprimir una hoja con las señales del osciloscopio debemos presionar la opción **Print** en el menú contextual. Lo que obtendremos es una hoja que contiene la pantalla del osciloscopio, con las señales presentes en ella y, debajo, veremos la información completa de la configuración del osciloscopio en ese momento. Si los cursores están activados, estos también se imprimirán.



OSCILOSCOPIO Y CARGA DE LA CPU

Como sabemos, cuanda la carga de la CPU es alta, las simulaciones no podrán correr en tiempo real. Si tenemos uno de estos casos y estamos usando el osciloscopio, al iniciar la simulación puede suceder que las señales tarden algunos segundos en aparecer en la pantalla. Esto quiere decir que cuanto más lenta sea la simulación, más tardarán en mostrarse.

Proyecto: control de velocidad de un motor DC por PWM

Como ejemplo de uso del osciloscopio, realizaremos la simulación de un sencillo circuito para controlar la velocidad de un motor de CD por modulación de ancho de pulso (PWM). Debemos descargar el archivo **PWM.dsn** para ver esta simulación.

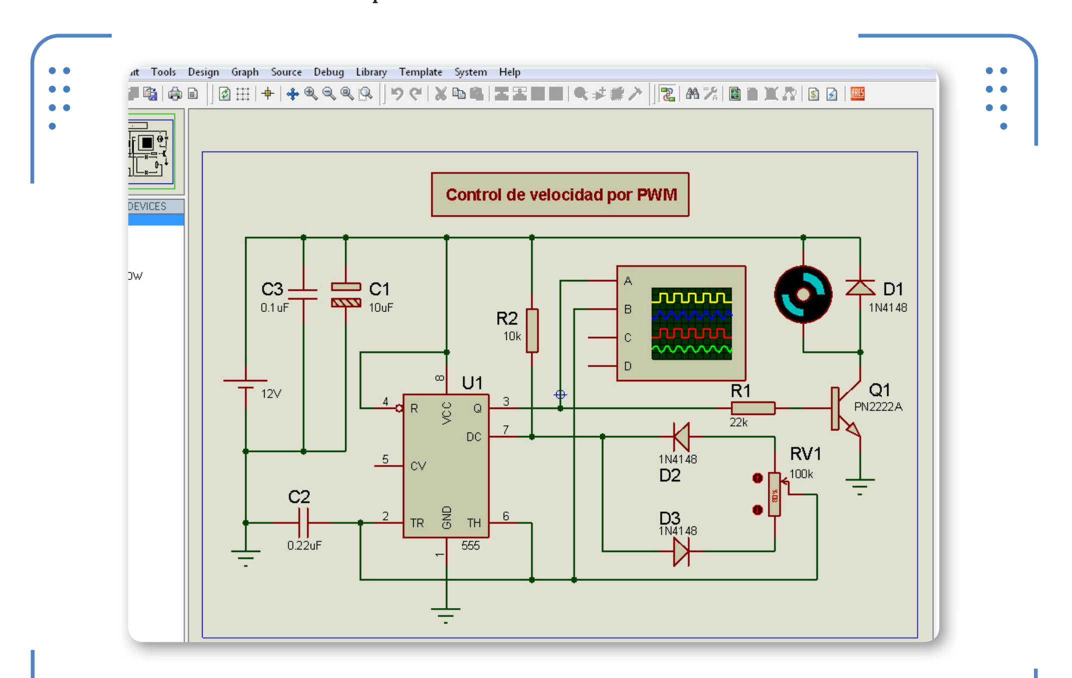


Figura 5. Un circuito PWM utilizando un 555 ilustra un ejemplo del uso del osciloscopio en ISIS.

En el circuito hemos incluido el osciloscopio; conectamos el **canal A** a la salida **Q** del 555, donde veremos la señal PWM, y el **canal B** a la terminal **TH** para observar la carga y descarga del capacitor **C2**. Al correr la simulación, aparecerá el osciloscopio con las señales en la pantalla. Si modificamos el valor de **RV1**, vemos el cambio en la velocidad del motor y en las señales. Durante la simulación, podemos cerrar la ventana del osciloscopio para ver el circuito con más detalle. Para recuperarlo, hacemos clic con el botón derecho del mouse sobre su símbolo, y en la parte inferior aparecerá la opción **Digital Oscilloscope**.



Contador/temporizador

El siguiente instrumento virtual que analizaremos es el **contador/ temporizador**, que será útil en las simulaciones digitales para medir tiempos, frecuencias o contar pulsos. Lo encontramos en la lista con el nombre **COUNTER TIMER**. Tal como los demás, lo agregamos al circuito como si fuera un componente y lo conectamos al punto donde queremos tomar la medición.

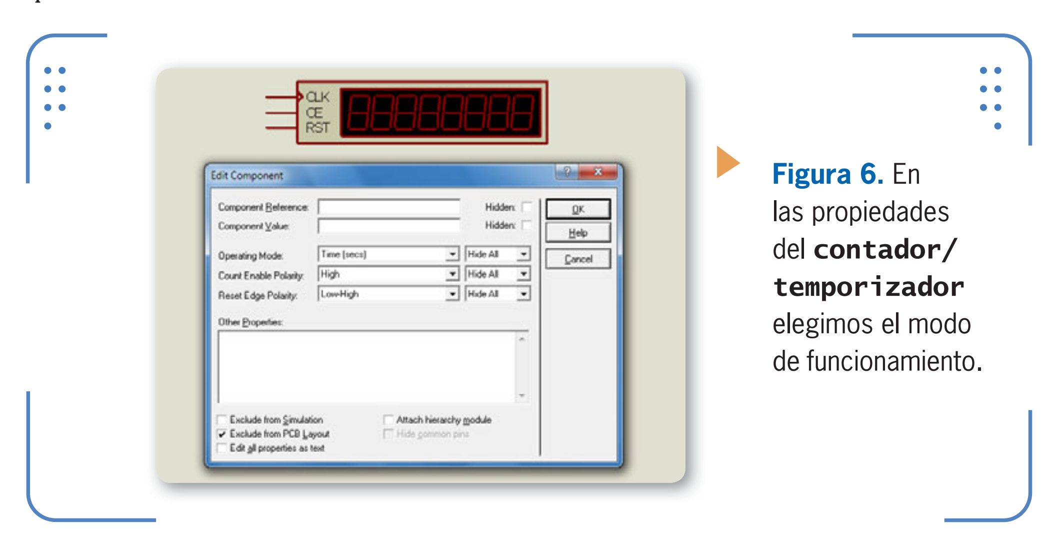


Figura 6. En las propiedades del **contador/ temporizador** elegimos el modo de funcionamiento.

El contador/temporizador tiene tres terminales de conexión: **CLK** es la que recibirá la señal para medir la frecuencia o contar pulsos, **CE** o **Count Enable** sirve para habilitar o deshabilitar el instrumento, y **RST** o **Reset**, para borrar y reiniciar la cuenta o medición. En las propiedades del contador/temporizador podemos configurar el modo y la función de los pines CE y RST. En el campo **Operating Mode** es posible elegir entre cuatro modos de operación: **Time (secs)** para medir el tiempo en segundos; **Time (hms)** para medir el tiempo en horas, minutos y segundos; **Frequency** para medir la frecuencia; y **Count** para contar los pulsos. El campo **Count Enable Polarity** permite seleccionar el nivel para habilitar o deshabilitar el instrumento

COUNTER/TIMER TIENE TRES TERMINALES DE CONEXIÓN: CLK, CE Y RST



con el pin CE; por ejemplo, si elegimos **High** (alto), el contador/temporizador estará habilitado con un estado alto y deshabilitado con un estado bajo. En el campo **Reset Edge Polarity** podemos establecer el flanco con el cual se dará el reset al instrumento: **High-Low** para un flanco de bajada o **Low-High** para un flanco de subida.

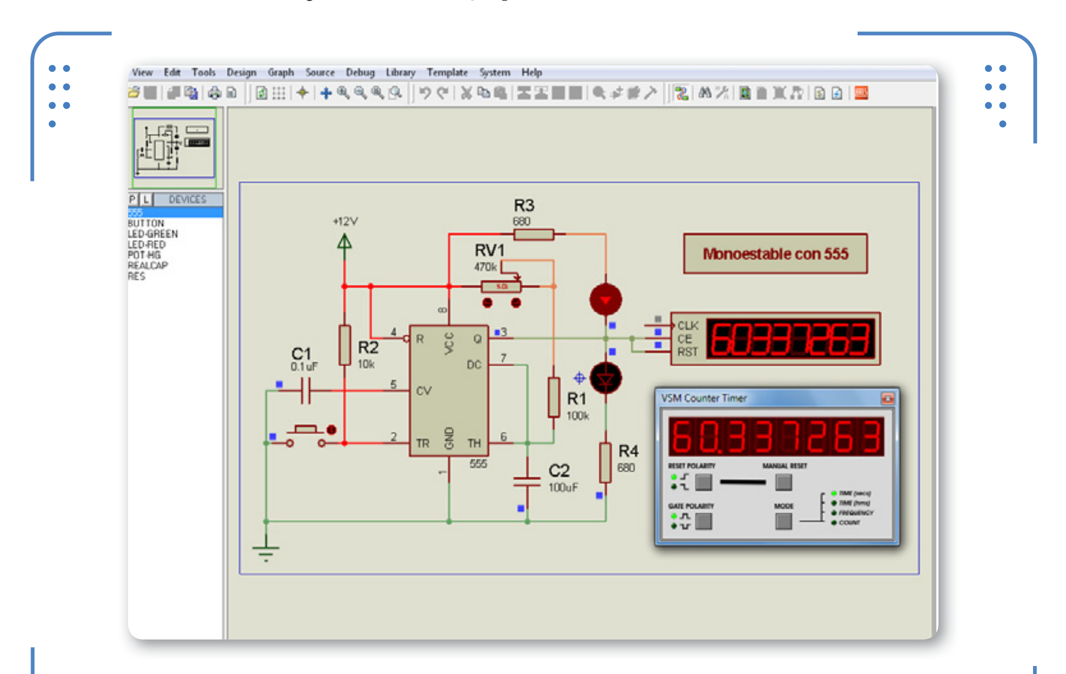


Figura 7. El **contador/temporizador** nos permite medir tiempos y otros parámetros en nuestras simulaciones.

Al correr la simulación, se mostrará la medición en el propio símbolo del contador/temporizador, pero si hacemos un clic sobre él, aparecerá

UNA MEJOR LECTURA DEL TEMPORIZADOR

En ocasiones, el símbolo del contador/temporizador puede ser muy pequeño (por ejemplo, en el caso de un circuito grande), además de que en la lectura no se muestra el punto decimal y esto puede ser confuso. Para una correcta visualización recomendamos siempre usar la ventana completa del instrumento, que aparece al hacer clic sobre él durante la simulación.

una ventana llamada **VSM Counter Timer**, que tiene diferentes botones para controlar su funcionamiento durante la simulación, incluyendo un botón de reset manual (**MANUAL RESET**).

Veamos un ejemplo del uso de este instrumento, para lo cual vamos a descargar el archivo **monoestable555.dsn**.

Este es un circuito monoestable construido con un 555, en el que, al presionar el pulsador, tendremos la salida **Q** en un estado alto por un tiempo determinado, que podemos variar usando el potenciómetro **RV1**. Hemos colocado un contador/temporizador en modo **Time (secs)** para medir el tiempo del pulso en el circuito. Las terminales CE y RST se han conectado a la salida Q del 555 y se ha elegido un nivel alto para habilitar el contador; de esta forma, al iniciar el pulso comenzará la cuenta del tiempo. La terminal CE se ha configurado para borrar el contador con un flanco de subida, de modo que, al iniciarse el pulso en Q, la cuenta empezará desde cero. Así podemos usar el contador/temporizador para medir de forma automática el tiempo del pulso de este monoestable.

VSM COUNTER TIMER PERMITE CONTROLAR EL CONTADOR DURANTE LA SIMULACIÓN





Generador de señales

Entre los instrumentos virtuales, encontramos el **SIGNAL GENERATOR**, que es capaz de generar señales senoidales, diente de sierra, triangulares y cuadradas, de una frecuencia de 0 Hz hasta 12 MHz y con una amplitud de hasta 12 Volts. Es completamente interactivo, ya que ni en su símbolo ni en sus propiedades podemos ver ni configurar nada.

GRÁFICOS DE SIMULACIÓN

El osciloscopio virtual de ISIS no es la única forma que tenemos para visualizar y analizar las señales analógicas en las simulaciones de circuitos. También tenemos la posibilidad de usar **gráficos de simulación**, que son un tipo de análisis no interactivo para poder ver las señales en los circuitos. De esto hablaremos ampliamente en el siguiente capítulo.

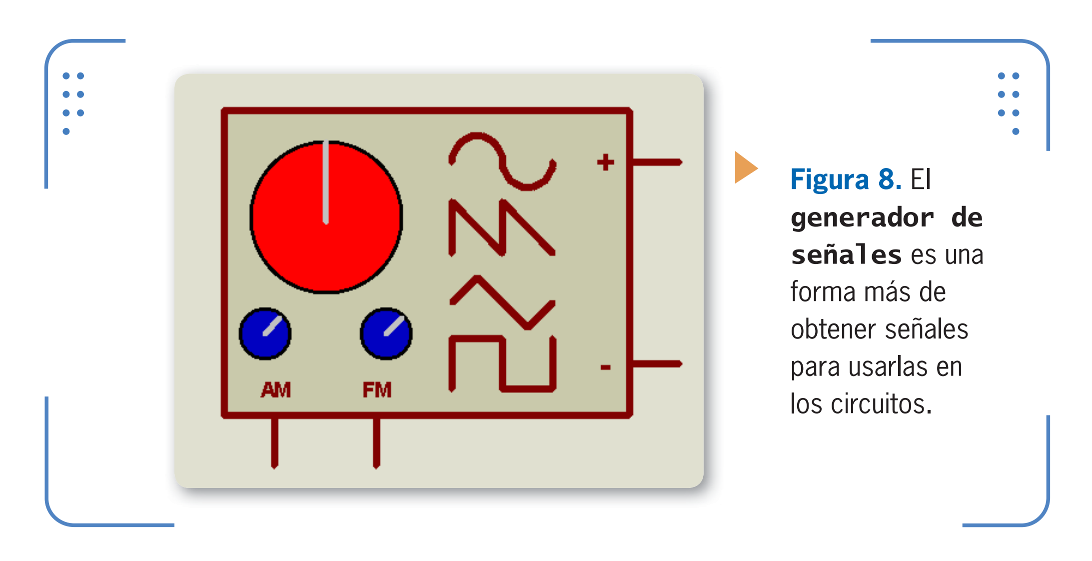
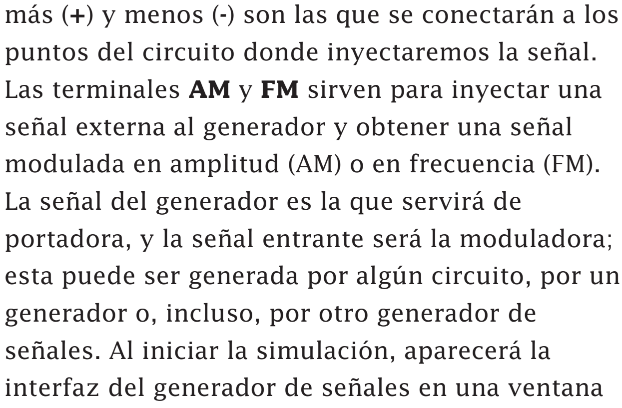


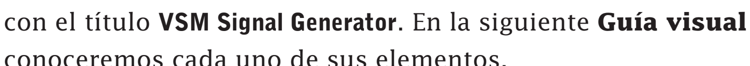
Figura 8. El **generador de señales** es una forma más de obtener señales para usarlas en los circuitos.

Como podemos apreciar en la **Figura 8**, el generador de señales es bastante simple: solo contiene cuatro terminales. Las marcadas con más (**+**) y menos (**-**) son las que se conectarán a los puntos del circuito donde inyectaremos la señal. Las terminales **AM** y **FM** sirven para inyectar una señal externa al generador y obtener una señal modulada en amplitud (AM) o en frecuencia (FM). La señal del generador es la que servirá de portadora, y la señal entrante será la moduladora; esta puede ser generada por algún circuito, por un generador o, incluso, por otro generador de señales. Al iniciar la simulación, aparecerá la interfaz del generador de señales en una ventana con el título **VSM Signal Generator**. En la siguiente **Guía visual** conoceremos cada uno de sus elementos.



EL GENERADOR DE SEÑALES ES MUY SIMPLE, SOLO CONTIENE CUATRO TERMINALES





GRÁFICOS DE SIMULACIÓN

Al igual que en el caso del osciloscopio virtual, el analizador lógico tampoco es la única manera de visualizar señales digitales. Es decir que también puede ser sustituido por los gráficos de simulación para el análisis de este tipo de señales en los circuitos. En el siguiente capítulo nos dedicaremos a estudiar en profundidad el tema de los **gráficos de simulación**.

GUÍA VISUAL ■ GENERADOR DE SEÑALES

01 **CENTRE/RANGE (CENTRO/RANGO):** este par de controles se usa para elegir la frecuencia de la señal que se va a generar.

02 **LEVEL/RANGE (NIVEL/RANGO):** con este par de controles definimos la amplitud de la señal.

03 **WAVEFORM (FORMA DE ONDA):** este botón permite cambiar la forma de onda generada, entre cuadrada, diente de sierra, triangular o senoidal.

04 **POLARITY (POLARIDAD):** mediante este botón, elegimos la polaridad. **Uni** para una señal unipolar, es decir, verticalmente solo estará en el lado positivo; y **Bi** para una señal bipolar, es decir, centrada en el eje horizontal, la mitad de la señal será negativa y la otra mitad positiva.

Figuras de Lissajous

Veamos un ejemplo práctico de los generadores de señales. Realizaremos una simulación con el archivo **Lissajous.dsn**, en el cual hemos utilizado un par de generadores de señales, para poder ver las **figuras de Lissajous** en el osciloscopio virtual en ISIS.



INSTRUMENTOS MEJORADOS

Algunos instrumentos virtuales se han mejorado en las últimas versiones de Proteus, principalmente, el osciloscopio virtual. El anterior solo era de dos canales, mientras que el más reciente es de cuatro. El analizador lógico es otro ejemplo: el anterior solo tenía 24 canales y el nuevo cuenta con 48.

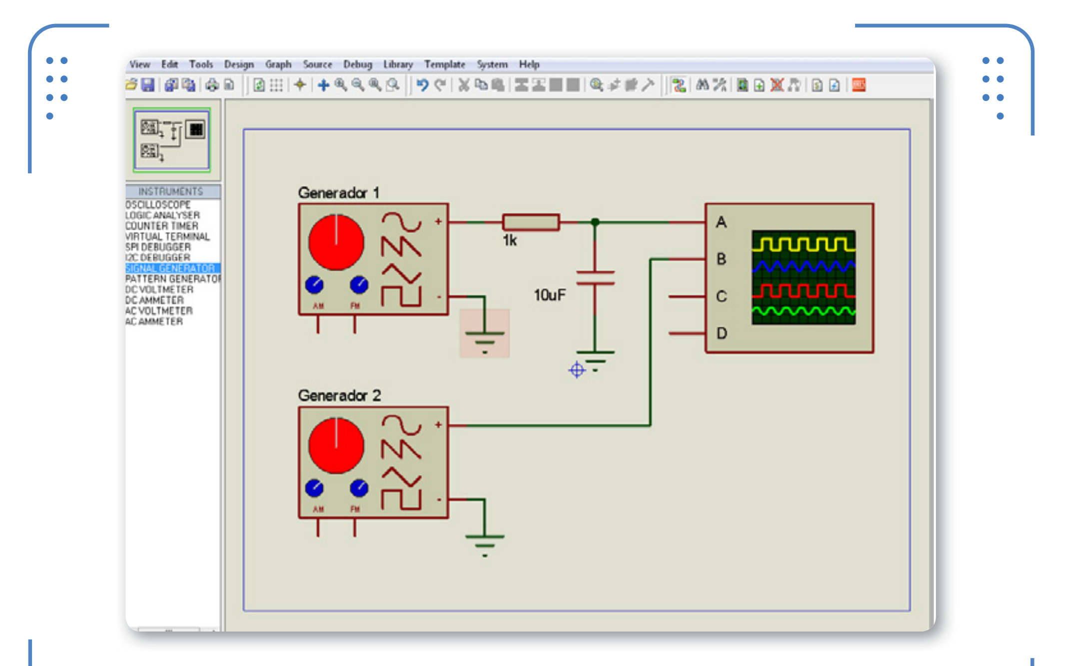


Figura 9. Podemos observar las conexiones de los generadores de señales para obtener las figuras de Lissajous.

Hemos colocado un filtro RC a la salida del primer generador de señales para obtener una señal con fase diferente de la segunda, y con el osciloscopio mediremos ambas señales. Al correr la simulación, podremos observar los generadores de señales y el osciloscopio con la figura de Lissajous.

Podemos cambiar la frecuencia de las señales para observar las variaciones en las figuras dibujadas en la pantalla del osciloscopio.



SEÑALES EN EL ANALIZADOR LÓGICO

Las señales correspondientes a las entradas A0 hasta A15 serán señales digitales individuales, mientras que si conectamos buses de 8 bits en las entradas B0 a B3, se mostrará una representación del momento de un cambio en el bus, cuando este ocurre, con líneas cruzadas verticales, y en hexadecimal podemos ver el valor que tiene el bus en cada momento. Debemos tener en cuenta que si los cambios en un bus se encuentran muy juntos los valores no se mostrarán.

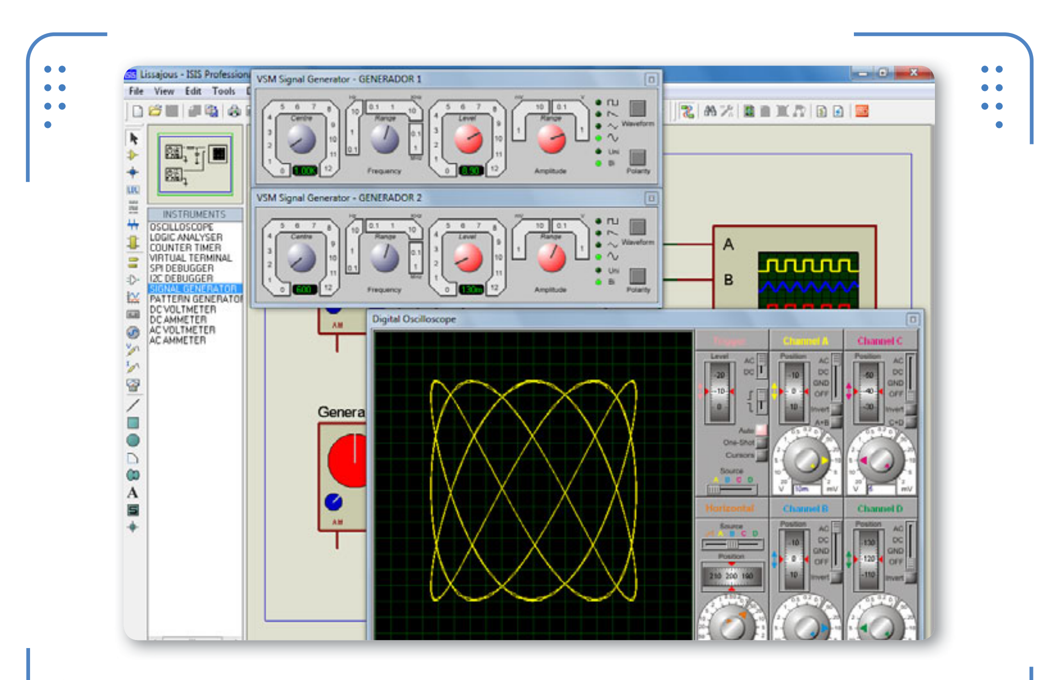


Figura 10. El osciloscopio virtual mostrará las figuras de Lissajous en su pantalla al iniciar la simulación.



Analizador lógico

Para poder analizar las señales en los circuitos digitales, ISIS dispone del **LOGIC ANALYZER**, que posee **48 canales**. Este analizador tiene 16 canales individuales y 4 analizadores de buses de 8 bits cada uno. Al igual que el generador de señales, es totalmente interactivo.

Los canales individuales (A0 hasta A15) pueden conectarse a señales individuales de los circuitos. Los segmentos gruesos de color azul indican la conexión de un bus de 8 líneas; en estas entradas es posible conectar, por ejemplo, un bus de datos o direcciones de una memoria para analizarlos. Los buses tienen una forma especial de conexión, que estudiaremos en el **Capítulo 6**. Una vez que tengamos la conexión del analizador lógico en el circuito, podemos correr la simulación, y la ventana del analizador lógico se mostrará con el nombre **VSM Logic Analyzer**. En la siguiente **Guía visual** conoceremos sus elementos.

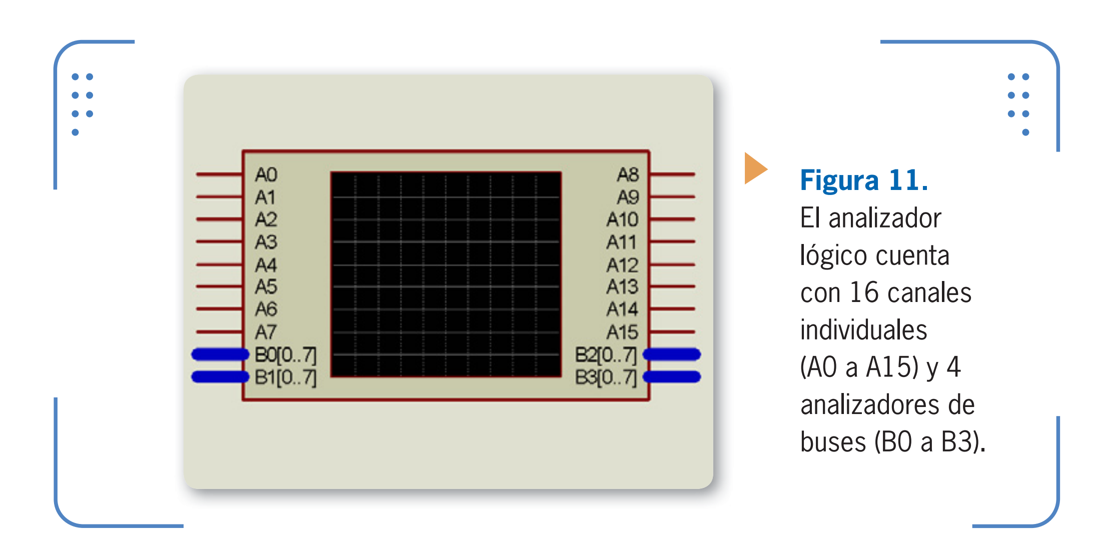
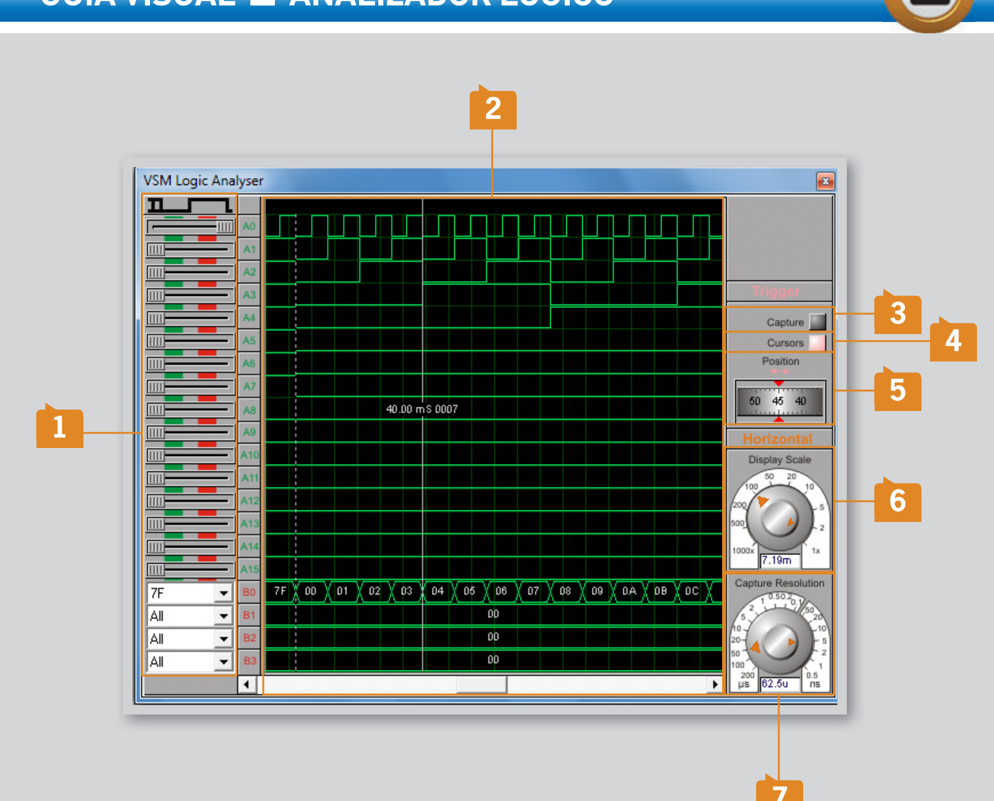


Figura 11. El analizador lógico cuenta con 16 canales individuales (A0 a A15) y 4 analizadores de buses (B0 a B3).

GUÍA VISUAL ■ ANALIZADOR LÓGICO



01	**TRIGGER CONDITIONS (CONDICIONES DE DISPARO):** en esta sección podemos definir las condiciones de disparo para los diferentes canales de entrada. En la parte superior vemos un gráfico que indica, de izquierda a derecha: cualquier estado o flanco, estado bajo (verde), flanco de subida, estado alto (rojo), flanco de bajada. En los buses (B0 a B3) se puede elegir el valor hexadecimal de la lista.
02	**SCREEN (PANTALLA):** aquí observaremos las señales.
03	**CAPTURE (CAPTURAR):** con este botón haremos una captura de las señales para mostrarlas en la pantalla. Al presionarlo, cambiará a color rosa; una vez que se logre la captura, cambiará a verde; finalmente, se apagará y se mostrarán las señales en la pantalla.
04	**CURSORS (CURSORES):** para activar o desactivar los cursores en la pantalla del analizador.
05	**POSITION (POSICIÓN):** permite cambiar la posición horizontal en la pantalla. El analizador tiene un buffer de 40000 muestras x 52 bits para cada captura; con este control se puede navegar a lo largo de todo el buffer tomando como referencia el 50% de este, que se indica con la línea punteada en la pantalla (**Marker**).
06	**DISPLAY SCALE (ESCALA):** se usa para ampliar o reducir la escala horizontal por división y, así, poder ver un área mayor o menor de las señales.
07	**CAPTURE RESOLUTION (RESOLUCIÓN DE CAPTURA):** para establecer la resolución con que se tomará la captura, es decir, el ancho mínimo de pulso que puede ser registrado por el analizador.

El analizador lógico también cuenta con la función de cursores para medir tiempos en la pantalla, de forma similar a los cursores del osciloscopio. Además, se pueden configurar los colores de la pantalla para el monitor o la impresión y, por supuesto, imprimir la pantalla con las señales en una hoja.

Análisis de un contador Johnson 4022

Veamos un ejemplo del uso del analizador lógico para comprender mejor su funcionamiento. Debemos descargar el archivo **AnalizadorLogico.dsn** para ver esta simulación. Tomaremos un circuito integrado 4022 para analizar su funcionamiento.

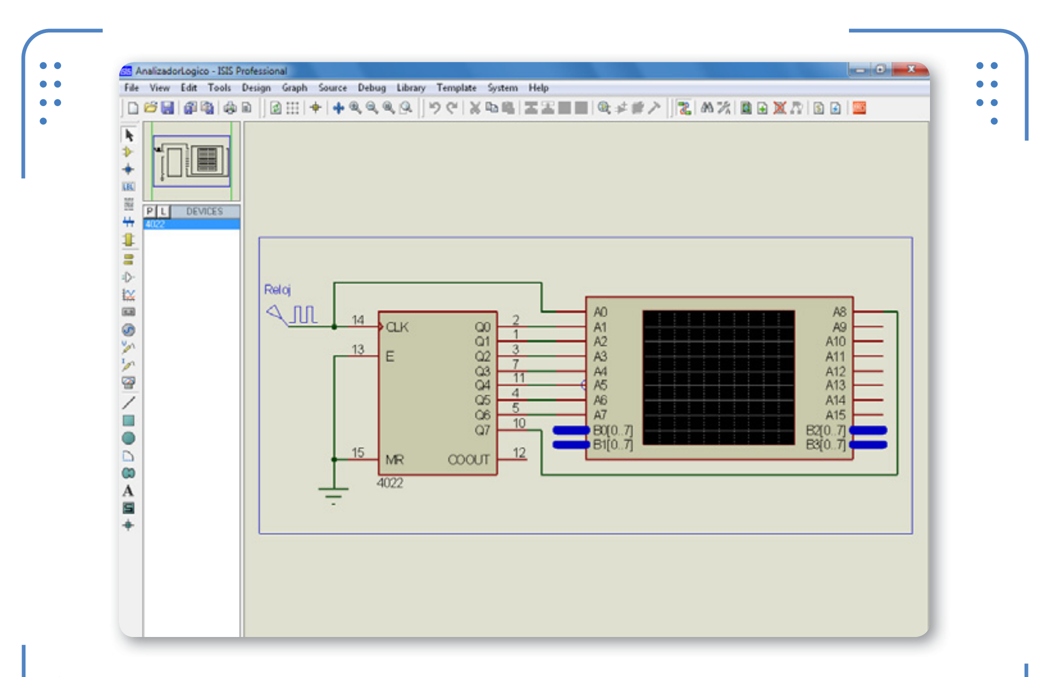


Figura 12. Un circuito integrado 4022 conectado al analizador lógico nos permite ver sus señales.

En este circuito hemos agregado un generador de reloj con una frecuencia de 1 KHz a la entrada de reloj del 4022. El analizador lógico se ha conectado de la siguiente forma: A0 a la señal de reloj, y A1 a A8 a las salidas del contador. Al correr la simulación, aparecerá la ventana del analizador y tendremos la siguiente configuración:

- La resolución de captura en 2 microsegundos.
- La escala en 0.5 milisegundos por división.
- La posición horizontal en 39 para poder ver la totalidad de las señales en la pantalla.
- El marcador (**Marker**) se ha desplazado a la izquierda de la pantalla.

En la sección de condiciones de disparo se ha configurado A0 en un nivel alto y A1 en un flanco de subida. Así, al tomar la captura, se habrá comenzando en el punto en donde se cumple esta condición, que es el inicio de la cuenta para este circuito integrado, y esta condición aparecerá justo en el marcador. Al presionar **Capture**, tomaremos la captura, y las señales se mostrarán en la pantalla del analizador lógico.

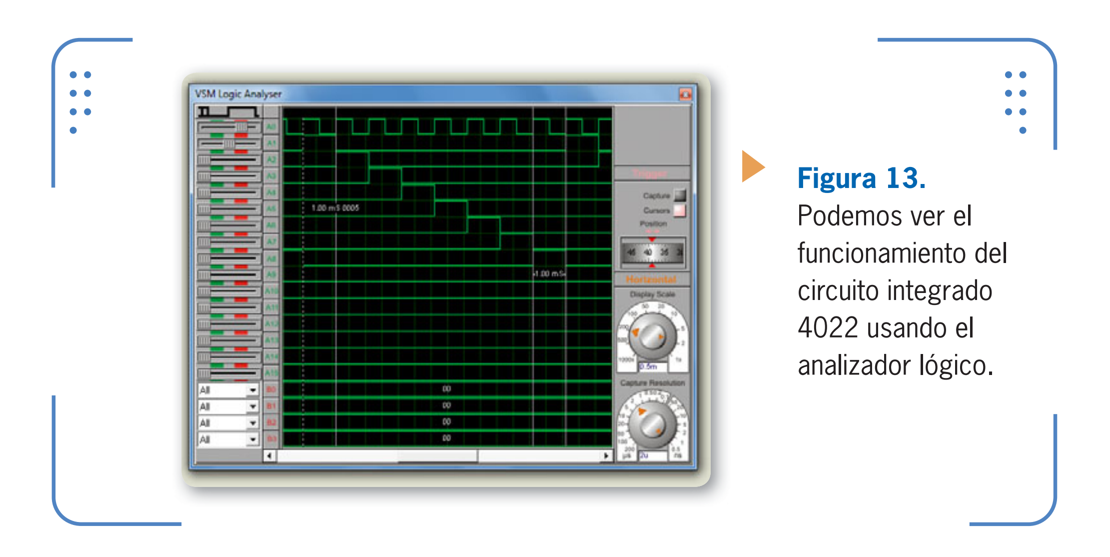


Figura 13. Podemos ver el funcionamiento del circuito integrado 4022 usando el analizador lógico.

Terminal virtual

Una **terminal virtual** sirve para transmitir o recibir datos de forma serial y puede usarse para verificar las transmisiones seriales en nuestros circuitos, ya sea recibiendo datos de la terminal virtual o enviando datos hacia ella para verificar que los reciba. En la lista de instrumentos virtuales aparece como **VIRTUAL TERMINAL** y usa el protocolo **RS232** para enviar o recibir datos.

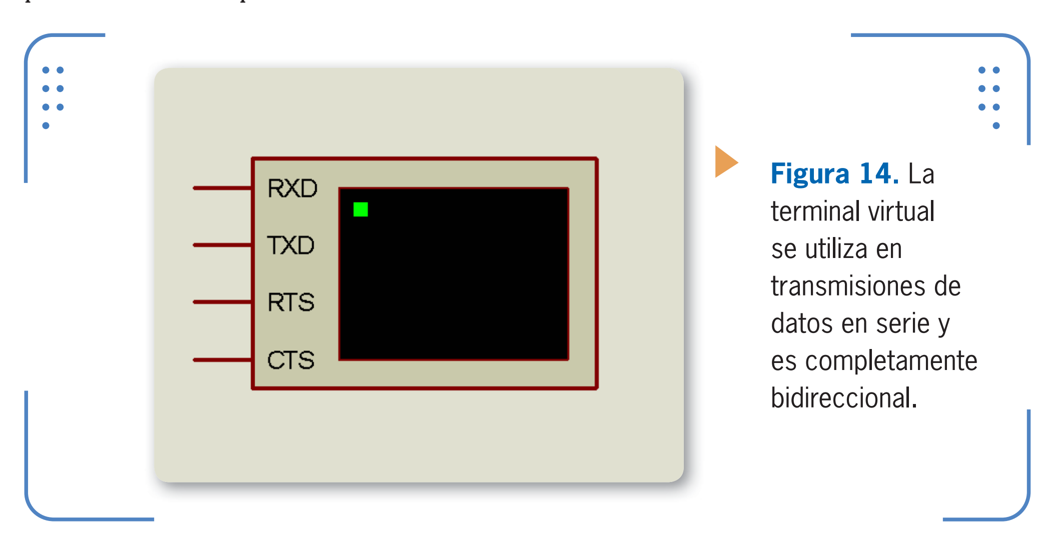


Figura 14. La terminal virtual se utiliza en transmisiones de datos en serie y es completamente bidireccional.

Las terminales de este instrumento son: **RXD** para recibir datos; **TXD** para enviar datos en formato **ASCII** desde el **teclado de la PC**, es decir, la terminal virtual enviará los datos ASCII que ingresemos hacia donde conectemos esta terminal; **RTS** (Ready to send) y **CTS** (Clear to send). En las propiedades de la terminal virtual podemos configurar los parámetros de la transmisión, incluyendo su velocidad.

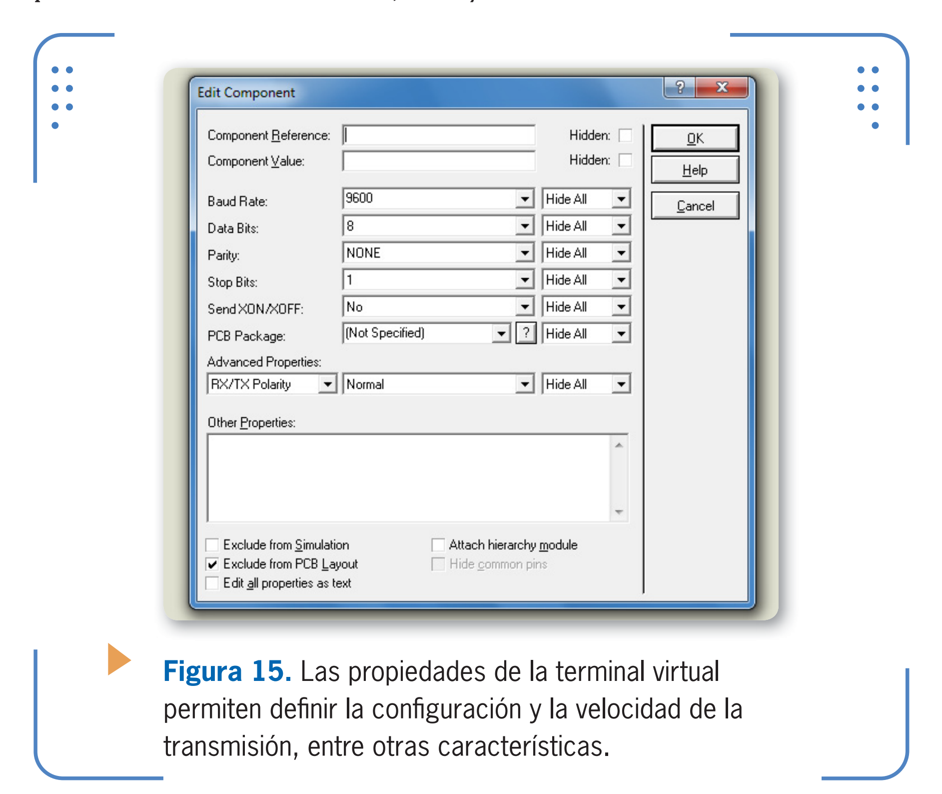


Figura 15. Las propiedades de la terminal virtual permiten definir la configuración y la velocidad de la transmisión, entre otras características.

Entre las configuraciones podemos definir:

- **Baud Rate**: este campo contiene la velocidad en baudios de la transmisión serial, puede ir de 110 a 57600 baudios.
- **Data Bits** (bits de datos): para indicar cuántos bits por dato se enviarán, las opciones son 7 u 8.
- **Parity** (paridad): aquí se define el bit de paridad; las opciones son **NONE** (ninguno), **EVEN** (par) u **ODD** (impar).
- **Stop Bits** (bits de detención): permite elegir los bits para la detención.
- **Send XON/XOFF**: en este campo debemos definir si se enviarán los comandos XON y XOFF o no.

- **Advanced Properties**: en este campo podemos establecer la polaridad de las señales en TXD/RXD y en RTS/CTS.

Figura 16. Una comunicación entre dos terminales virtuales realizada con éxito en la simulación.

La forma más fácil de ejemplificar el uso de las terminales virtuales es generando una comunicación entre dos de ellas, para lo cual, simplemente, conectamos la terminal TXD de una a la terminal RXD de la otra. Esto podemos verlo en el archivo **TerminalVirtual.dsn**. Al iniciar la simulación, se mostrarán dos ventanas llamadas **Virtual Terminal – TRANSMISOR** y **Virtual Terminal – RECEPTOR**. Si hacemos un clic en la



ENSEÑAR CON INSTRUMENTOS VIRTUALES

Los instrumentos virtuales de Proteus pueden ser sumamente útiles para enseñar a los principiantes su utilización, sobre todo si no se cuenta con estos instrumentos en la realidad, ya que algunos de ellos son muy costosos. Además no hay daños ante posibles errores en su uso, estos instrumentos son irromplibles.

ventana del transmisor para resaltarla, podremos escribir un mensaje mediante el teclado de nuestra computadora. Los datos se transmitirán hacia el receptor y se mostrarán en su ventana. De esta manera hemos realizado una comunicación serial entre las dos terminales virtuales.

EN LA VENTANA DEL TRANSMISOR PODEMOS ESCRIBIR UN MENSAJE CON EL TECLADO DE LA PC

Las dos terminales deben estar configuradas de forma idéntica para que la transmisión se lleve a cabo sin fallas; el nombre que dimos a las terminales nos permite identificarlas en la simulación. Si hacemos un clic con el botón derecho sobre la ventana de una terminal, se abrirá un menú contextual que contiene algunas opciones de configuración. Encontramos: borrar la pantalla (**Clear Screen**), pausar la transmisión (**Pause**), copiar o pegar (**Copy/Paste**), hacer que se reflejen en la pantalla los caracteres que escribimos (**Echo Typed Characters**), cambiar a modo hexadecimal (**Hex Display Mode**) o modificar la fuente que se mostrará en la ventana de la terminal (**Set Font**).

Textos automáticos al inicio de la simulación

Podemos especificar una cadena de texto que se enviará de forma automática al iniciar la simulación en una terminal virtual. Para hacerlo, en las propiedades de la terminal virtual debemos escribir, por ejemplo, en el campo **Other Properties**: **TEXT=Hola** o **{TEXT=Hola}** (las llaves se usan para que este atributo esté oculto). Esto enviará automáticamente el texto **Hola** al iniciar la simulación a través de la terminal TXD de esa terminal virtual.





TERMINALES VIRTUALES VS. MAX232

A pesar de que Proteus cuenta con un modelo del circuito integrado **MAX232**, debemos darle prioridad al uso de terminales virtuales, ya que el modelo de estas es puramente digital, lo cual genera simulaciones mucho más ligeras. El modelo del MAX232 es analógico y produce una importante carga para la CPU.



Generador de patrones digitales

Otro instrumento de gran utilidad es el **generador de patrones digitales**, que sirve para generar series de datos binarios en paralelo de forma personalizada. Se pueden generar patrones simples o complejos con una capacidad de hasta 1k x 8, es decir, hasta 1024 datos de 8 bits cada uno. En la siguiente **Guía visual** conoceremos sus elementos.

▼ GUÍA VISUAL ■ GENERADOR DE PATRONES

01 **CLKIN (ENTRADA DE RELOJ):** esta terminal sirve para conectar una señal de reloj en caso de usarse en modo de reloj externo.

02 **B[0..7]:** bus de salida de datos.

03 **Q0 A Q7:** terminales de salida de datos.

04 **CASCADE (CASCADA):** esta terminal es usada para conectar otros generadores de patrones en cascada. Se pone en estado alto durante la salida del primer dato de la secuencia y permanece en estado bajo durante el resto del tiempo.

05 **CLKOUT (SALIDA DE RELOJ):** es la salida de la señal de reloj interno.

06 **OE (HABILITAR SALIDA):** se usa para habilitar la salida de datos o inhabilitarla, poniendo las salidas en estado de alta impedancia. La salida se habilita con un estado alto en ella y se deshabilita con un estado bajo; el generador continuará funcionando aun si las salidas están deshabilitadas.

07 **TRIG (DISPARO):** se utiliza para dar un disparo externo al generador y que este inicie su funcionamiento.

08 **HOLD (PAUSA):** permite detener o pausar el funcionamiento del generador. Se detendrá en el punto donde se encuentre al habilitar este pin con un estado alto, y continuará al regresar al estado bajo.

Es posible configurar el generador de patrones para que funcione con reloj interno o externo, además de poder manejar su funcionamiento mediante los pines de control (**HOLD**, **TRIG**, **OE**). La salida de datos del generador será en las terminales **Q0** a **Q7**, en el bus de datos **B[0..7]**, o en ambos.

El generador de patrones puede ser configurado ya sea mediante su ventana de propiedades o también de manera interactiva, es decir, durante la simulación. En la siguiente **Guía visual** conoceremos en detalle las propiedades del generador y los parámetros que podemos modificar en ellas antes de correr la simulación.

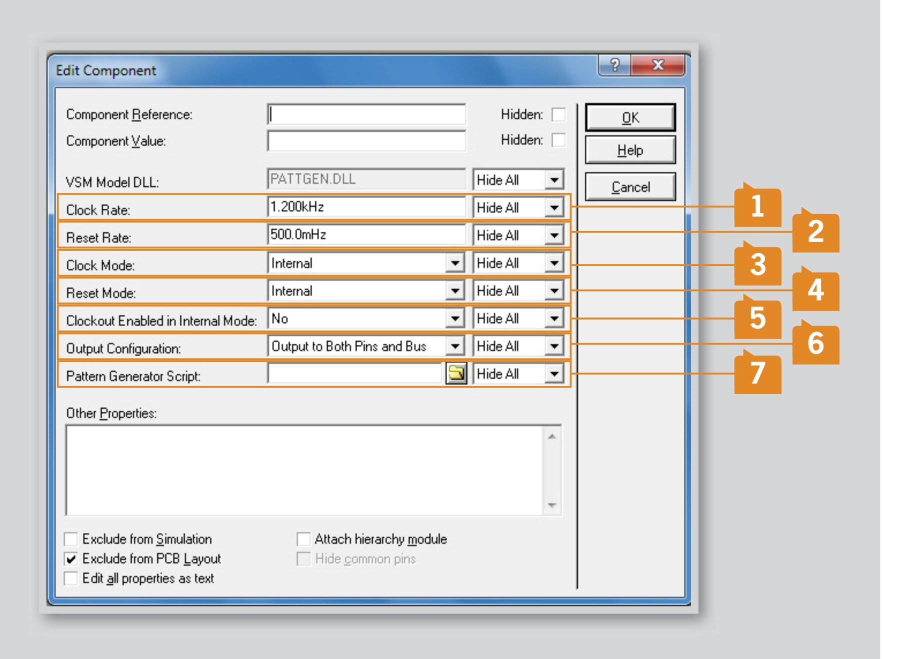


GENERAR PATRONES SERIALES

El generador de patrones es una herramienta diseñada para las secuencias de datos en paralelo. Sin embargo, también es posible usarlo para generar patrones seriales, simplemente, empleando y configurando una de las salidas (por ejemplo, Q0) e ignorando las demás, o generando varias secuencias seriales simultáneas en más de una salida.

GUÍA VISUAL ■ PROPIEDADES DEL GENERADOR





01 **CLOCK RATE (FRECUENCIA DE RELOJ):** en este campo definimos la frecuencia de reloj en caso de usarse el reloj interno del generador de patrones.

02 **RESET RATE (FRECUENCIA DE DISPARO):** aquí indicamos la frecuencia con que el generador de patrones reiniciará su funcionamiento, si se ha elegido el disparo interno.

03 **CLOCK MODE (MODO DE RELOJ):** en esta lista seleccionamos el modo del reloj: **Internal** para usar el reloj interno, **External Pos Edge** para un reloj externo con flancos de subida o **External Neg Edge** para un reloj externo con flancos de bajada.

04 **RESET MODE (MODO DE DISPARO):** en esta lista elegimos el modo de disparo entre las siguientes opciones: **Internal** para disparo interno; **Async External Pos Edge** para disparo externo asíncrono con flancos de subida; **Sync External Pos Edge** para disparo externo síncrono con flancos de subida; **Async External Neg Edge** para disparo externo asíncrono con flancos de bajada, y **Sync External Neg Edge** para disparo externo síncrono con flancos de bajada.

05 **CLOCKOUT ENABLED IN INTERNAL MODE (SALIDA DE RELOJ EN MODO INTERNO):** acá podemos indicar si la señal de reloj interna se reflejará en la salida CLKOUT o no.

06 **OUTPUT CONFIGURATION (CONFIGURACIÓN DE SALIDA):** permite establecer en qué salidas tendremos señales: **Output to Both Pins and Bus**, para tener salidas en el bus B[0..7] y en las terminales Q0 a Q7 al mismo tiempo; **Output to Pins Only**, para salidas en los pines Q0 a Q7 solamente, y **Output to Bus Only** para salida solo a través del bus B[0..7].

07 **PATTERN GENERATOR SCRIPT (ARCHIVO DE PATRONES):** en este campo podemos elegir un archivo de patrones para cargarlo en el generador; los archivos de patrones tienen la extensión **.PTN**.

Al correr la simulación, aparecerá la ventana del generador de patrones **VSM Pattern Generator**, en donde es posible configurar los parámetros y definir el patrón por generar. Si es la primera vez que usamos el generador de patrones en una simulación, podemos presionar el botón **Pause** (pausa) en la barra de simulación; así se iniciará la simulación en pausa y podremos configurar el patrón o los parámetros del generador.

▼ GUÍA VISUAL ■ VENTANA DEL GENERADOR

01	**VENTANA DE PATRÓN:** la cuadrícula representa el patrón por generar. Los cuadros blancos son ceros, y los negros, unos. Cada columna representa la salida de un dato, y de arriba hacia abajo corresponden a las salidas Q0 a Q7 (o B0 a B7 para el bus). En la parte inferior de cada columna aparece el valor representado en ella en hexadecimal, en la parte superior se numeran las columnas, y la pequeña flecha con la línea punteada representa el fin del patrón.
02	**FRECUENCIA DE DISPARO INTERNO:** en esta sección tenemos los controles para el disparo interno. Con **Trigger** controlamos el rango y con **Vernier**, el ajuste fino.
03	**FRECUENCIA DE RELOJ INTERNO:** aquí ajustaremos la frecuencia del reloj interno del generador. Es importante resaltar que el reloj interno siempre trabajará con flancos de bajada.
04	**MODO DE DISPARO:** en esta sección tenemos botones para configurar el disparo, ya sea interno o externo, y si será síncrono o asíncrono.
05	**STEP (PASO):** permite probar el generador paso a paso. Cada vez que lo presionemos, la simulación correrá hasta que el siguiente patrón se refleje en la salida, y entrará en pausa hasta ser presionado otra vez.
06	**MODO DE RELOJ:** para configurar el modo de reloj, interno o externo.

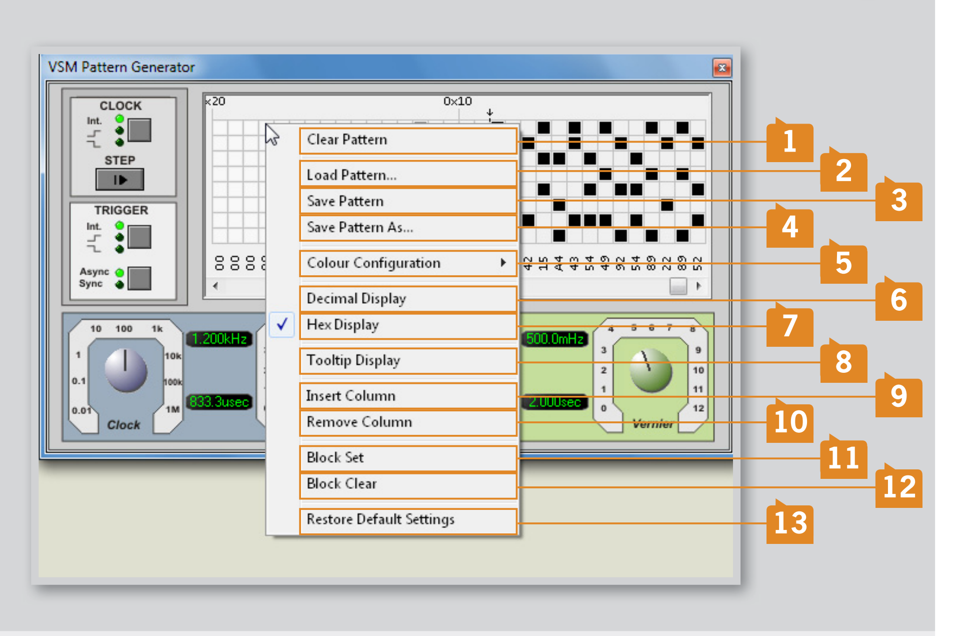
Al usar las perillas para cambiar la frecuencia de disparo o del reloj, puede ser difícil lograr los valores deseados. En aquellos casos en que necesitemos valores exactos, deberemos hacer un doble clic sobre alguna de las perillas; entonces aparecerá una pequeña ventana de texto, donde ingresaremos el valor requerido.

El patrón se define cambiando los valores en la cuadrícula de la ventana de patrón. Simplemente, debemos hacer un clic para indicar si es un **uno** o un **cero**. También podemos dar un clic sobre el valor de la parte inferior de cada columna y editarlo directamente ingresando un nuevo número. Es posible desplazar la flecha que indica el fin del patrón hacia la posición deseada o hacer un clic en la parte superior de la cuadrícula para mostrarla si no está visible.

Si hacemos un clic con el botón derecho del mouse sobre la ventana del patrón, aparecerá un menú contextual con algunas opciones útiles. En la siguiente **Guía visual**, las conoceremos en detalle.

GUÍA VISUAL ■ MENÚ CONTEXTUAL





01 **CLEAR PATTERN (BORRAR PATRÓN):** para borrar completamente el patrón.

02 **LOAD PATTERN (CARGAR PATRÓN):** permite cargar un archivo de patrones **.PTN**; al seleccionar esta opción, aparecerá el navegador para buscar el archivo en el sistema.

03 **SAVE PATTERN (GUARDAR PATRÓN):** guarda el patrón actual.

04 **SAVE PATTERN AS (GUARDAR PATRÓN COMO):** permite guardar el patrón actual eligiendo la carpeta de destino y el nombre del archivo **.PTN**.

05 **COLOUR CONFIGURATION (CONFIGURACIÓN DE COLORES):** contiene varias opciones para cambiar la configuración de los colores en la ventana de patrón.

06 **DECIMAL DISPLAY (MOSTRAR EN DECIMAL):** al elegir esta opción, los valores en la parte inferior y superior de la cuadrícula en la ventana de patrón se mostrarán en decimal.

07 **HEX DISPLAY (MOSTRAR EN HEXADECIMAL):** al seleccionar esta opción, los valores en la parte inferior y superior de la cuadrícula en la ventana de patrón se mostrarán en hexadecimal.

08	**TOOLTIP DISPLAY (MOSTRAR SUGERENCIAS):** activa las sugerencias; se muestra un cuadro de color amarillo al lado del cursor del mouse con información de las columnas y filas mientras se desplaza por el patrón.
09	**INSERT COLUMN (INSERTAR COLUMNA):** inserta una nueva columna en el lugar donde hicimos el clic derecho. Las columnas del lado izquierdo se desplazan hacia la izquierda, y la última columna se borra al insertar una nueva.
10	**REMOVE COLUMN (BORRAR COLUMNA):** borra la columna sobre la cual hicimos el clic derecho. Las columnas del lado izquierdo se desplazan hacia la derecha, y una nueva columna se agrega al final.
11	**BLOCK SET (PONER BLOQUE A UNOS):** aparecerá un cursor en forma de cruz en la ventana de patrón. Al hacer un clic en algún lugar, podremos arrastrar para formar un cuadro de selección. Cuando soltamos, todos los cuadros elegidos se pondrán a uno.
12	**BLOCK CLEAR (PONER BLOQUE A CEROS):** similar a la opción anterior, solo que en esta la selección de cuadros que hagamos se pondrá a cero.
13	**RESTORE DEFAULT SETTINGS (RESTAURAR CONFIGURACIÓN POR DEFECTO):** permite restaurar los colores por defecto en la ventana de patrón, y los valores se mostrarán en hexadecimal, que es la opción predeterminada.

Hemos visto una gran cantidad de opciones de configuración del generador de patrones. Para finalizar con el estudio de esta herramienta, la pondremos en funcionamiento. Podemos ver un sencillo ejemplo del uso del generador de patrones mediante el archivo **GeneradorPatrones.dsn**, en el que hemos colocado un circuito **4543** y un **display**. Usaremos el generador de patrones para dar una secuencia de 0 a 9 y mostrar la cuenta en el display.



DIFERENCIA ENTRE DISPAROS

El disparo interno es por intervalos definidos, sin importar la señal de reloj. El síncrono está en sincronía con el flanco del reloj, y el asíncrono se da en el momento en que ocurre, sin importar el reloj. Por ejemplo, si elegimos disparo externo síncrono, al recibirse la señal de disparo, el generador esperará hasta que se dé el siguiente flanco correcto de la señal de reloj.

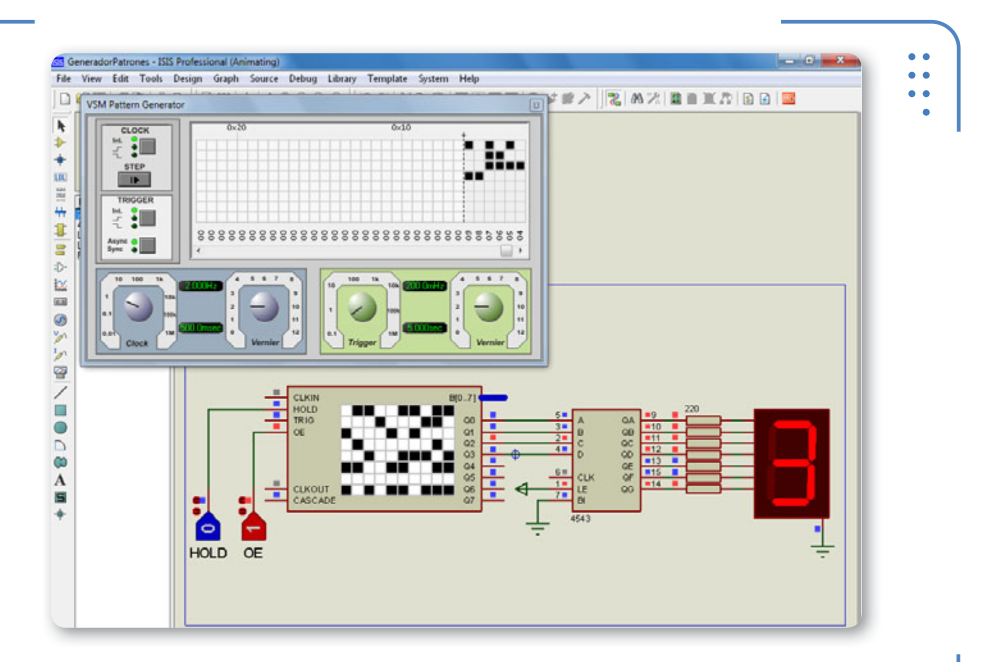


Figura 17. Un contador BCD usando el generador de patrones es un ejemplo muy simple de su funcionamiento.

Hemos configurado el generador con una frecuencia de reloj interno de 2 Hz y una frecuencia de disparo interno de 0.2 Hz, para que la cuenta inicie inmediatamente al finalizar la anterior. No debemos olvidar que es importante calcular la frecuencia de disparo para no interrumpir la cuenta antes de que se complete.

Con este ejemplo pudimos comprobar el funcionamiento del generador de patrones, solo nos queda utilizarlo en nuestro trabajo.



PARÁMETROS INSTANTÁNEOS

En el **Capítulo 2** observamos cómo, durante una simulación en pausa, podemos acceder a los **parámetros instantáneos** de los componentes. Para los instrumentos virtuales también está disponible esta opción, excepto para el contador/temporizador y para el analizador lógico. En este último aparecerá un mensaje que dice que son demasiados pines para mostrarlos.



Los archivos PWI y los instrumentos virtuales

En el **Capítulo 1** mencionamos que, al guardar un diseño en Proteus, pueden generarse otros archivos además del **.DSN**. En caso de utilizar instrumentos virtuales, y en especial los que se abren en una ventana independiente, como el osciloscopio, el contador/temporizador, el generador de señales, el analizador lógico, la terminal virtual y el generador de patrones, se creará el archivo **.PWI** cuando almacenemos el diseño. Este guarda la posición de la ventana, su tamaño (en los casos en que se puede redimensionar) y, además, si la ventana del instrumento virtual está visible al correr la simulación o no.

Si borramos los archivos **.PWI** no afectaremos al archivo **.DSN**, pero la posición, el tamaño y la visibilidad de los instrumentos virtuales que contiene serán revertidos a los valores por defecto. Entonces, si queremos conservarlos como estaban al guardar el diseño, tendremos que almacenar el archivo **.PWI** siempre en la misma carpeta donde se encuentra el **.DSN** y con el mismo nombre.



RESUMEN

Así como en nuestro laboratorio de electrónica contamos con múltiples instrumentos de medición, en ISIS tenemos todo un laboratorio virtual a nuestro alcance. En este capítulo hemos estudiado cuáles son los instrumentos virtuales y cómo se utilizan. La similitud con los instrumentos reales es intencional, ya que de esta manera podremos usar Proteus como una herramienta de aprendizaje o, simplemente, para tomar mediciones, como si lo hiciéramos en la realidad. Hemos omitido los analizadores de comunicación SPI e I2C, ya que los estudiaremos en el **Capítulo 6**.

Actividades

TEST DE AUTOEVALUACIÓN

1. ¿Cómo se puede elegir un instrumento virtual en ISIS?
2. ¿Cómo se deben conectar los amperímetros en los circuitos?
3. ¿Qué parámetros podemos medir con los cursores del osciloscopio virtual?
4. Si durante la simulación cerramos la ventana de algún instrumento virtual, ¿cómo podemos abrirla de nuevo?
5. ¿Con qué instrumento virtual podemos medir el tiempo?
6. ¿Con qué instrumento virtual podemos generar una onda triangular?
7. ¿Cuántos canales tiene el analizador lógico de Proteus?
8. ¿Para qué sirve una terminal virtual?
9. ¿Para qué sirve el generador de patrones?
10. El generador de patrones puede funcionar con un reloj interno o _______.

ACTIVIDADES PRÁCTICAS

1. Abra el archivo **PWM.dsn**, agregue un contador/temporizador y configúrelo en modo Frequency.
2. Conecte la terminal CLK a la salida Q del 555, inicie la simulación y observe la frecuencia de la señal medida.
3. Abra el archivo **Monoestable555.dsn** y determine el tiempo mínimo y máximo del pulso de salida, colocando RV1 a sus valores mínimo y máximo.
4. En un nuevo diseño, coloque un circuito integrado 4024, un generador DCLOCK y visualice su funcionamiento con el analizador lógico de la misma forma en que lo hicimos con el 4022.
5. Abra el archivo **AnalizadorLogico.dsn**, cambie la frecuencia del generador DCLOCK a 100 Hz, corra la simulación y ajuste los parámetros del analizador lógico para ver las señales de manera clara en él.



Análisis con gráficos

Hasta ahora hemos visto simulaciones interactivas, pero existe otra manera de simulación no interactiva que puede resultar sumamente útil en muchos circuitos.
En este capítulo estudiaremos los gráficos de simulación, los cuales permiten realizar un poderoso análisis de los circuitos mediante la visualización de diferentes tipos de señales analógicas y digitales.

Análisis mediante gráficos de simulación

En el capítulo anterior vimos algunos instrumentos de medición para analizar circuitos de modo interactivo en la ventana de ISIS, pero esa no es la única manera de medir señales: también existe otra poderosa herramienta, que son los **gráficos de simulación**. Estos gráficos constituyen una forma no interactiva de analizar los parámetros de los circuitos. Para acceder a los que se encuentran disponibles en ISIS, existe un botón en la barra de herramientas de **Modo** llamado **Graph Mode**, que tiene la forma, precisamente, de un pequeño gráfico. Al presionarlo, entraremos en **Modo de gráficos** y en el **Selector de objetos** se mostrará la lista de los gráficos disponibles.

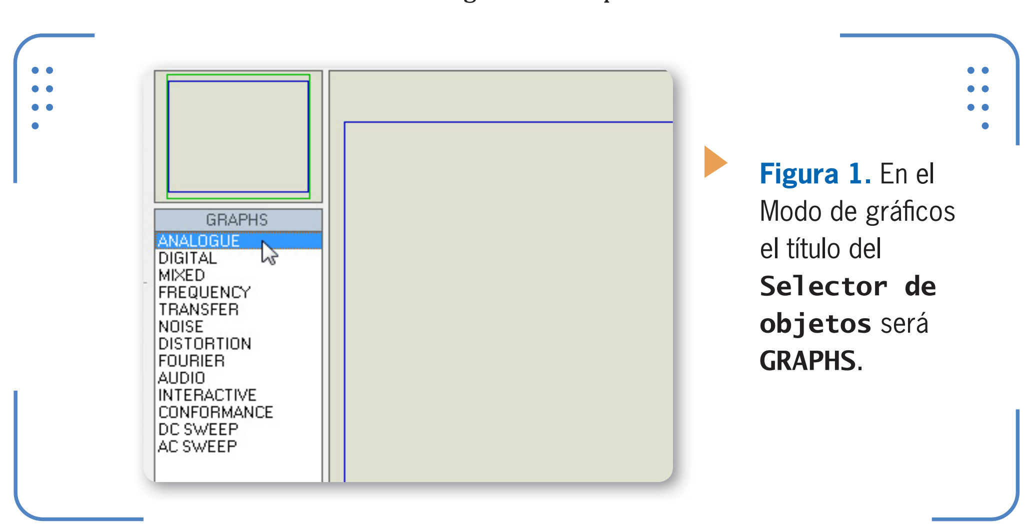


Figura 1. En el Modo de gráficos el título del **Selector de objetos** será **GRAPHS**.



ANÁLISIS NO TEMPORAL

El osciloscopio y el analizador lógico virtuales de Proteus realizan un análisis temporal de las señales en los circuitos, es decir, en el dominio del tiempo. En cambio, con algunos gráficos de simulación se puede hacer un análisis no temporal, por ejemplo, en el dominio de la frecuencia, o incluso en el dominio de parámetros personalizados como resistencia, voltaje, temperatura, etcétera.



Gráfico analógico

Los gráficos se utilizan para **medir señales** en algún punto del circuito, definido por una sonda de voltaje o corriente o por un generador. Tomaremos un ejemplo real de una simulación para aprender paso a paso cómo crear un gráfico de simulación; para hacerlo, debemos descargar el archivo **AmplificadorInversor.dsn**.

El primer gráfico que estudiaremos es el analógico o **ANALOGUE**, que como su nombre bien lo indica, es el que se utiliza para realizar el análisis de señales analógicas.

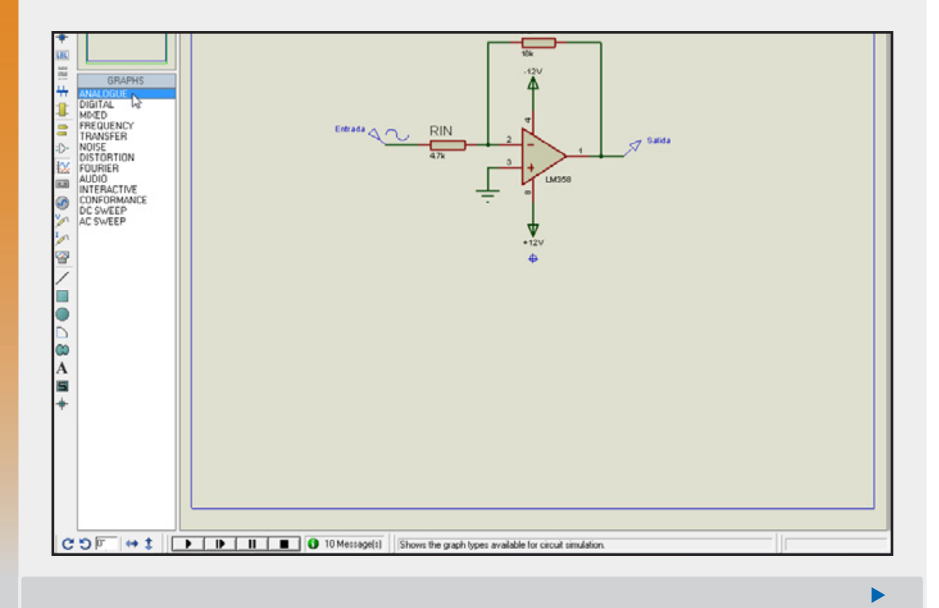
LOS GRÁFICOS MIDEN SEÑALES EN UN PUNTO DEL CIRCUITO DEFINIDO POR UNA SONDA O GENERADOR



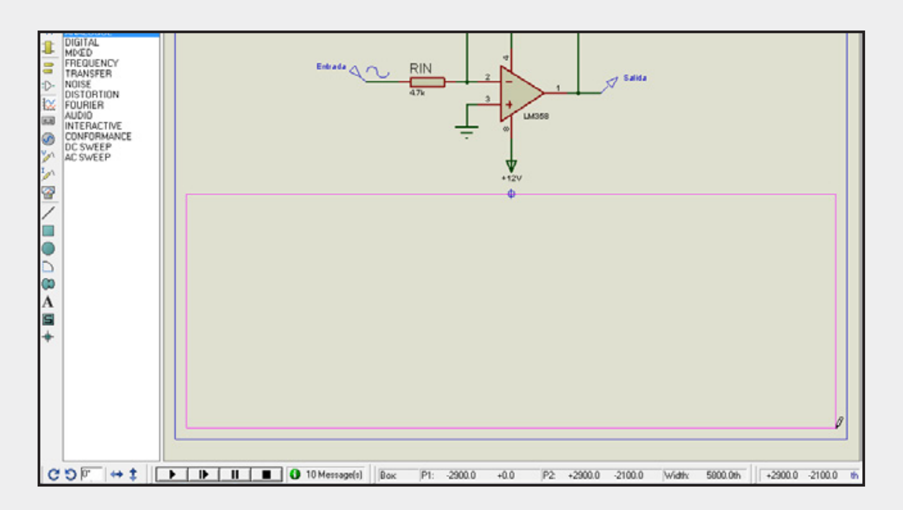
PASO A PASO: CREAR UN GRÁFICO DE SIMULACIÓN



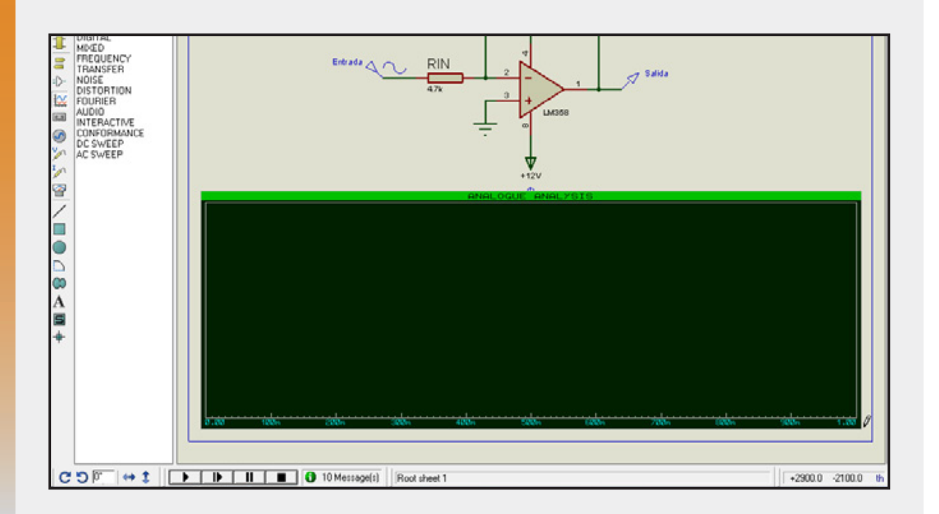
01 Abra el archivo `AmplificadorInversor.dsn`, luego pulse el botón `Graph Mode` y seleccione el primer gráfico de la lista, llamado ANALOGUE.



02 Lleve el cursor del mouse a la ventana de Edición y note cómo toma la forma de un lápiz de color blanco. Haga un clic en algún lugar y arrastre para formar un cuadro de color rosa; no es necesario mantener el botón del mouse presionado. Si necesita cancelar esta operación, haga un clic con el botón derecho del mouse o presione la tecla ESC.



03 Cuando esté satisfecho con el tamaño y la posición del cuadro, haga un clic; aparecerá la ventana ANALOGUE ANALYSIS, que es el gráfico de simulación analógico.



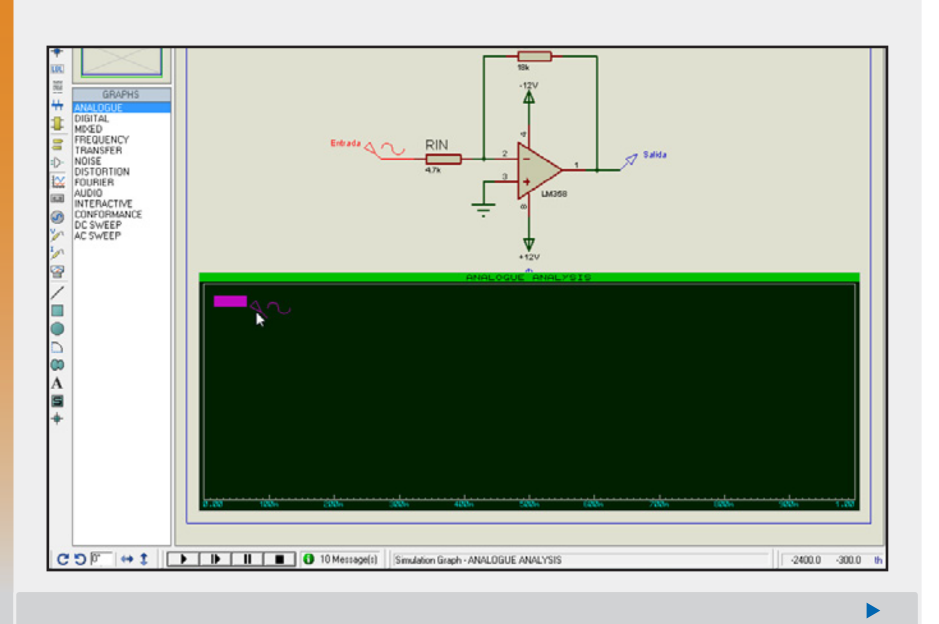
De esta manera, hemos creado un gráfico de simulación analógico en nuestro diseño. Es posible generar el gráfico en cualquier lugar de la ventana de trabajo y con cualquier tamaño. Por supuesto, una vez creado, podemos moverlo, seleccionándolo y arrastrándolo, o también redimensionarlo, si lo preferimos.

Por ahora el gráfico está vacío, y para analizar alguna señal debemos indicarle en qué punto la tomaremos. Con este fin, vamos a usar un generador o una sonda de voltaje o corriente. Como podemos ver en el archivo de ejemplo, tenemos un simple amplificador inversor construido con un LM358, un generador llamado **Entrada** que provee una señal y, además, una sonda de voltaje llamada **Salida** en la que tendremos la señal amplificada. Debemos indicarle al gráfico creado qué señales analizaremos y en qué puntos del circuito; hay varias formas de hacerlo. Veamos el siguiente **Paso a paso**.

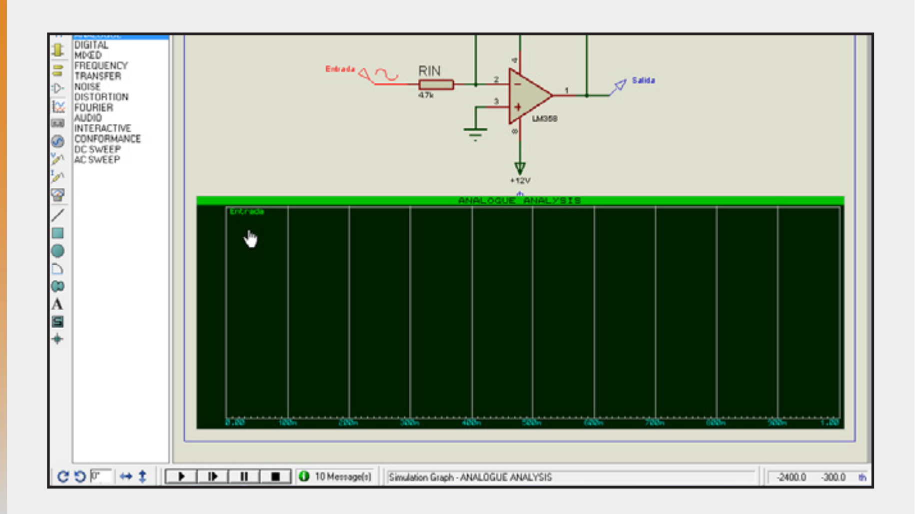
PASO A PASO: AGREGAR UNA SEÑAL AL GRÁFICO



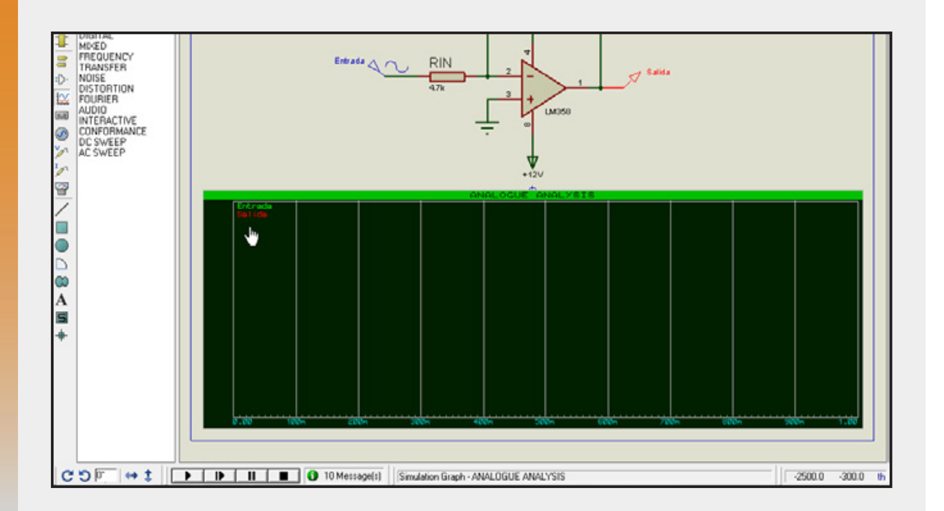
01 En el circuito de ejemplo `AmplificadorInversor.dsn` con el gráfico ya agregado, haga un clic en el generador llamado Entrada para seleccionarlo y arrástrelo hacia la parte superior izquierda del gráfico.



02 Suelte el botón del mouse, y el nombre del generador aparecerá en color verde dentro de la ventana del gráfico. También verá algunas líneas de división en él, lo que indica que ha agregado un punto de prueba al gráfico.



03 Realice el mismo procedimiento con la sonda de voltaje llamada Salida, seleccionándola y arrastrándola hacia el gráfico. El nombre de la sonda aparecerá debajo del nombre del generador y esta vez su color será rojo.



Así hemos agregado los puntos del circuito donde el gráfico hará el análisis de las señales. Es posible agregar tantos puntos como sean necesarios, y cada uno tendrá un color diferente en el gráfico.

Generar el gráfico de simulación

Una vez que tenemos listas las señales para medir en el gráfico correspondiente, ya podemos analizarlas. Para esto, simplemente presionamos la **Barra espaciadora** en nuestro teclado, con lo cual el gráfico se calculará y aparecerán las señales en él. También podemos recurrir al menú **Graph/Simulate Graph**.

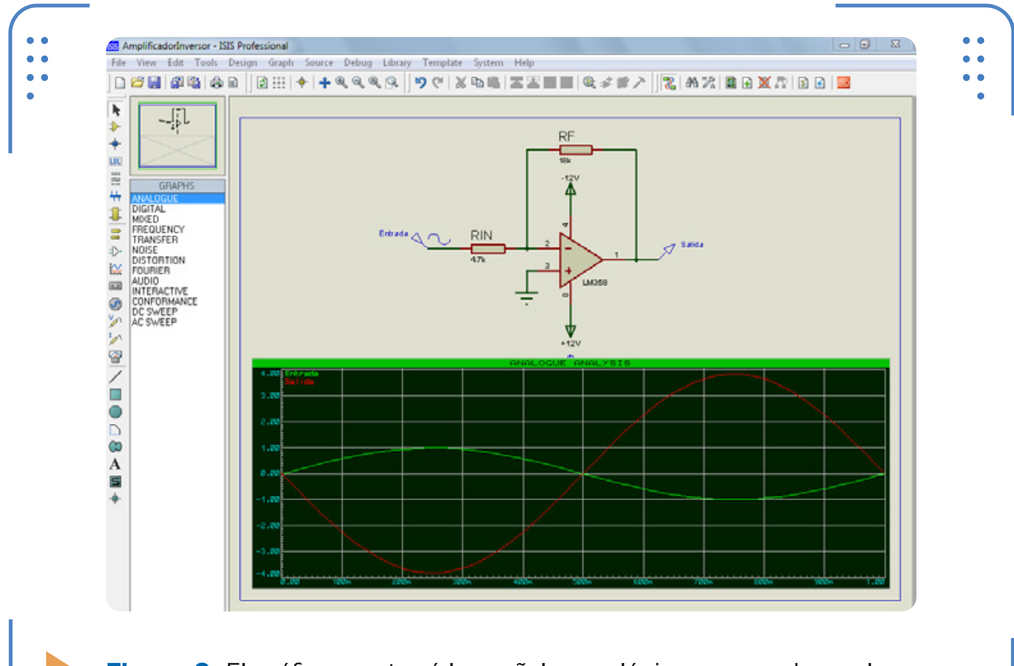
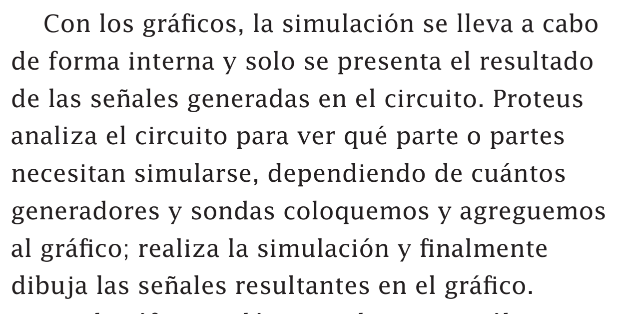


Figura 2. El gráfico mostrará las señales analógicas generadas en los puntos elegidos en el circuito.

En este ejemplo, al simular el gráfico obtendremos las señales correspondientes a los puntos que hemos elegido. La señal en color verde es la del generador, y la señal en color rojo es la salida del amplificador; el eje Y (vertical) del gráfico muestra la **amplitud** de las señales, y el eje X, (horizontal) el **tiempo**. La escala del eje Y se ajusta

automáticamente para mostrar las señales con claridad. Al simular el gráfico, vemos que el proceso no es interactivo, es decir, no se muestra el funcionamiento del circuito como en las simulaciones de los capítulos anteriores.



EN LOS GRÁFICOS LA SIMULACIÓN SE REALIZA DE FORMA INTERNA: NO SON INTERACTIVOS

Con los gráficos, la simulación se lleva a cabo de forma interna y solo se presenta el resultado de las señales generadas en el circuito. Proteus analiza el circuito para ver qué parte o partes necesitan simularse, dependiendo de cuántos generadores y sondas coloquemos y agreguemos al gráfico; realiza la simulación y finalmente dibuja las señales resultantes en el gráfico.

En el gráfico analógico se hace un análisis temporal, es decir, en el dominio del tiempo, en semejanza a lo que hace un osciloscopio. Otros gráficos realizan un análisis diferente, como veremos más adelante.

Gráficos con doble eje vertical

Imaginemos que en el circuito de nuestro amplificador queremos graficar la corriente de entrada. Para hacerlo, podemos incorporar una sonda de corriente inmediatamente después del generador, y agregar esta nueva sonda al gráfico; aparecerá su nombre en color azul debajo de las dos señales anteriores. Al calcular el gráfico otra vez, presionando la **Barra espaciadora**, veremos que la línea azul que representa la corriente es prácticamente una recta en cero. Esto se debe a que la escala de corriente es muy diferente de las demás y no podemos apreciarla bien. Para solucionar este problema, podemos hacer uso del eje vertical del lado derecho del gráfico. Hasta ahora hemos agregado las señales al gráfico arrastrándolas hacia el lado



ALTERNATIVAS DE ANÁLISIS

Los gráficos de simulación son una alternativa al análisis que se puede hacer con el osciloscopio, el analizador lógico e, incluso, los voltímetros o amperímetros. Aunque los gráficos no son interactivos, pueden resultar útiles para un análisis más detallado de las señales generadas en los circuitos.

izquierdo, pero si arrastramos una sonda o generador hacia el eje derecho, se ubicará de ese lado del gráfico. Así generaremos gráficos con escalas verticales diferentes

En este ejemplo, hemos agregado la sonda de corriente en el lado derecho del gráfico, pero como en ese eje se muestra una escala distinta, podemos apreciar la señal de corriente (azul), junto con los gráficos que representan voltaje. Las señales agregadas del lado derecho del gráfico muestran su nombre en la parte inferior.

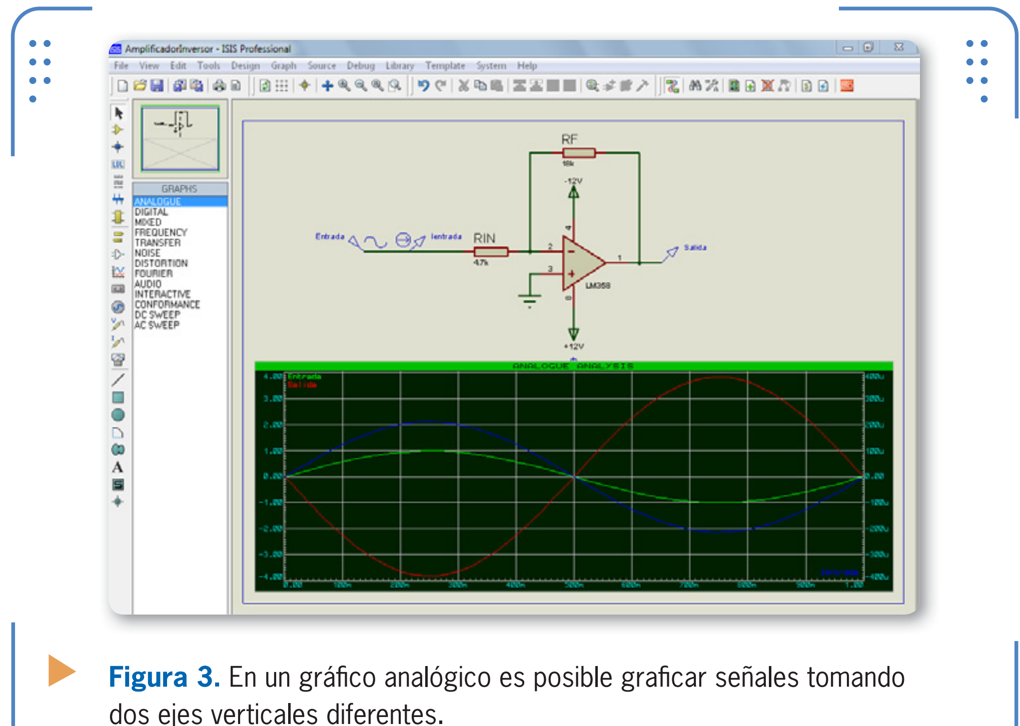


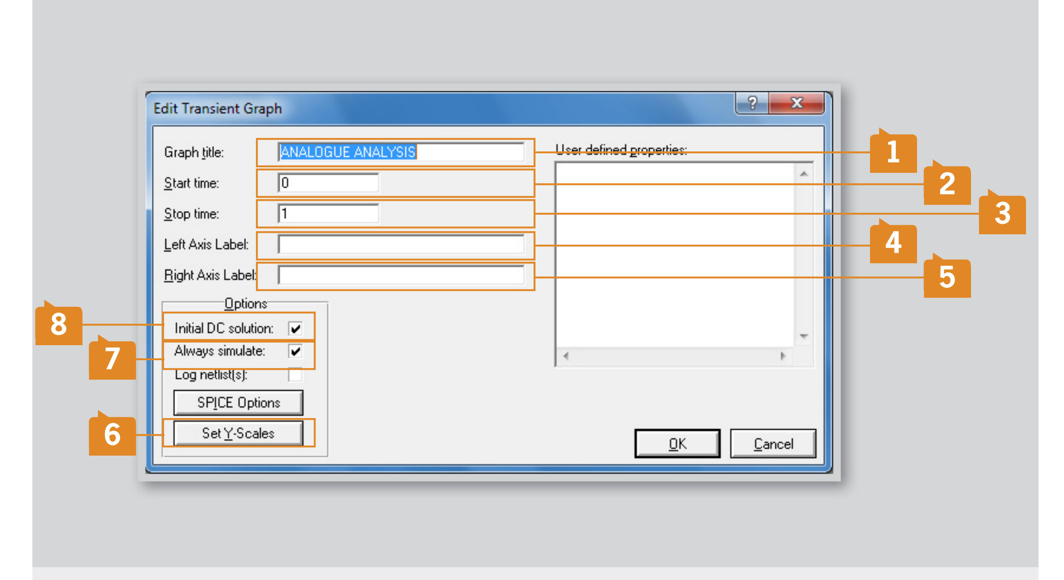
Figura 3. En un gráfico analógico es posible graficar señales tomando dos ejes verticales diferentes.

Propiedades del gráfico

Los gráficos de simulación también tienen propiedades que podemos editar, tal como lo hicimos en otros elementos. Accedemos a ellas desde el menú contextual que se abre al pulsar el botón derecho del mouse o haciendo un doble clic sobre el gráfico. Aparecerá una ventana con el título **Edit Transient Graph**. En la siguiente **Guía visual** explicaremos sus propiedades.

GUÍA VISUAL ■ PROPIEDADES DEL GRÁFICO





01 **GRAPH TITLE (TÍTULO DEL GRÁFICO):** este campo define el título del gráfico (que aparece arriba en la barra verde de este); por defecto siempre será **ANALOGUE ANALYSIS** para el gráfico analógico, pero podemos colocar el título que necesitemos.

02 **START TIME (TIEMPO INICIAL):** el eje horizontal representa el tiempo en este gráfico, donde podemos especificar el tiempo de inicio del gráfico.

03 **STOP TIME (TIEMPO FINAL):** aquí se indica el tiempo final, es decir, el tiempo máximo del eje horizontal.

04 **LEFT AXIS LABEL (ETIQUETA DEL EJE IZQUIERDO):** permite ingresar un nombre o etiqueta para el eje vertical del lado izquierdo, si lo necesitamos. El nombre aparecerá arriba de las señales agregadas en el gráfico.

05 **RIGHT AXIS LABEL (ETIQUETA DEL EJE DERECHO):** en este campo podemos definir un nombre o etiqueta para el eje vertical derecho.

06 **SET Y-SCALES (AJUSTAR EJES VERTICALES):** este botón permite ajustar las escalas de los ejes verticales manualmente.

07 **ALWAYS SIMULATE (SIEMPRE SIMULAR):** si marcamos esta opción, el gráfico siempre será simulado. Si la desmarcamos, al intentar calcular el gráfico, ISIS hará una comprobación, y si el circuito no ha cambiado con respecto al último cálculo, no se simulará. ▶



INITIAL DC SOLUTION (CONDICIONES INICIALES): especifica si serán utilizadas condiciones iniciales en el circuito o no, especialmente para los inductores y capacitores. Si la opción está marcada, al calcular el gráfico, todos los capacitores estarán descargados y todos los inductores tendrán un flujo igual a cero en el tiempo cero de la simulación.

Las escalas de los ejes verticales o ejes Y se ajustan de forma automática al calcular el gráfico; si las señales cambian, las escalas también se ajustarán a las nuevas señales. Usando el botón **Set Y-Scales**, podemos fijar los valores con las opciones **Lock Values?**, y especificar un mínimo y un máximo manualmente, si lo necesitamos. Esto puede hacerse en todos los tipos de gráficos.

Ampliar la ventana del gráfico

Podemos ampliar la ventana de un gráfico para verla con más detalle o acceder a otras funciones. Para hacerlo, debemos realizar un clic con el botón derecho del mouse sobre el gráfico y, en el menú contextual que se abre, elegir la opción **Maximise (Show Window)**, o simplemente podemos hacer un **doble clic** en la barra de título, que es de color verde y contiene el nombre del gráfico (o un solo clic, si el gráfico está seleccionado).

En la siguiente **Guía visual** vamos a explicar en detalle cada uno de sus elementos para lograr una correcta configuración de ellos.

TENEMOS DIFERENTES MANERAS DE AMPLIAR LA VENTANA DEL GRÁFICO





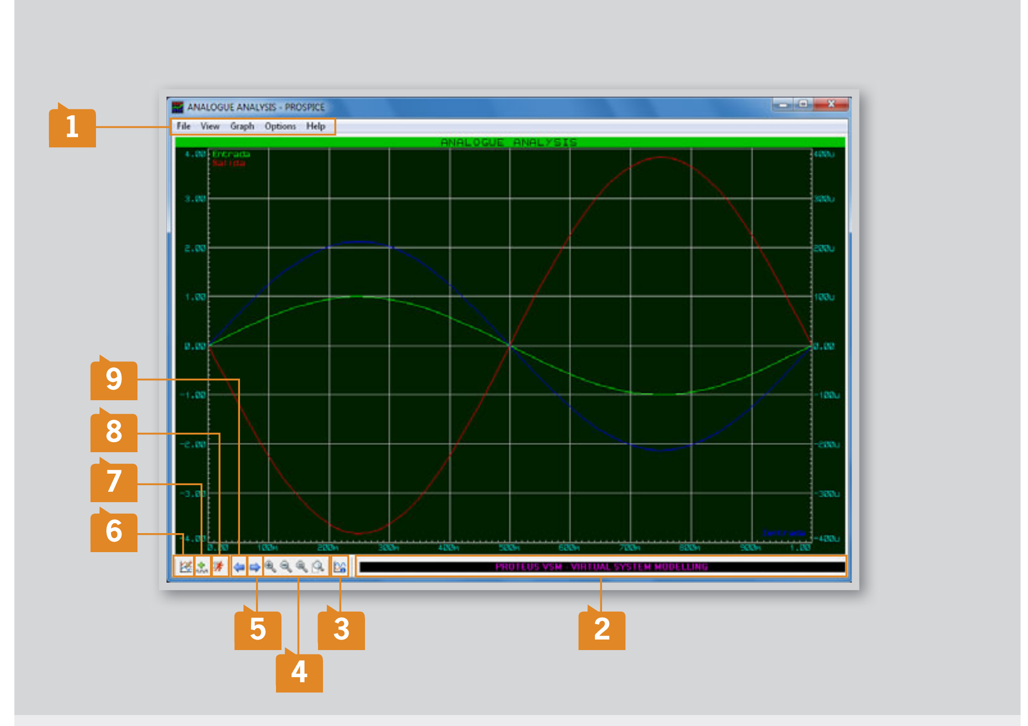


LIMPIAR UN GRÁFICO

Debemos saber que después de calcular un gráfico, las señales permanecerán dibujadas en él. Incluso si guardamos el diseño, al abrirlo otra vez, ahí estarán. Entonces, si deseamos borrar las señales dibujadas en un gráfico, debemos ir al menú **Graph** y luego seleccionar la opción **Clear Data** para dejar el gráfico vacío hasta que necesitemos calcularlo de nuevo.

▼ GUÍA VISUAL ■ VENTANA AMPLIADA DE UN GRÁFICO





01 **BARRA DE MENÚS:** en esta sección encontramos los menús que contienen opciones para el gráfico, a los que también se pueden acceder desde la ventana de ISIS

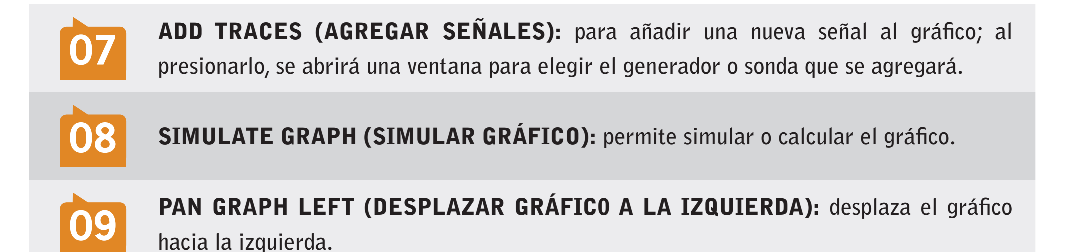
02 **BARRA DE ESTADO DEL GRÁFICO:** esta barra presenta información relevante sobre las señales en el gráfico, como amplitudes o tiempos al usar los cursores.

03 **VIEW SIMULATION LOG (VER REPORTE DE SIMULACIÓN):** este botón muestra la ventana con el reporte de simulación.

04 **BOTONES DE ZOOM:** funcionan de manera idéntica a los botones de la barra de herramientas de visualización, pero aplicados al gráfico.

05 **PAN GRAPH RIGHT (DESPLAZAR GRÁFICO A LA DERECHA):** desplaza el gráfico hacia la derecha.

06 **EDIT GRAPH (PROPIEDADES DEL GRÁFICO):** abre la ventana de propiedades del gráfico. ▶



07 **ADD TRACES (AGREGAR SEÑALES):** para añadir una nueva señal al gráfico; al presionarlo, se abrirá una ventana para elegir el generador o sonda que se agregará.

08 **SIMULATE GRAPH (SIMULAR GRÁFICO):** permite simular o calcular el gráfico.

09 **PAN GRAPH LEFT (DESPLAZAR GRÁFICO A LA IZQUIERDA):** desplaza el gráfico hacia la izquierda.

La ventana expandida del gráfico aparecerá como una ventana independiente en el sistema. Podemos moverla, redimensionarla o incluso maximizarla para verla en toda la pantalla. Si regresamos a la ventana de ISIS mientras un gráfico está expandido, en su lugar habrá un aviso que nos lo advierte.

Es posible analizar las señales en el gráfico expandido. Si hacemos un clic sobre una de las señales o cerca de ella, aparecerá un cursor vertical de color verde con una pequeña cruz en el punto donde cruza la señal seleccionada. En la barra de estado del gráfico se mostrarán el tiempo y la amplitud correspondientes a dicho punto, y el nombre de la señal seleccionada. Podemos desplazar el cursor horizontalmente para seguir la señal y ver su amplitud en cada punto. Si queremos elegir otra señal para analizarla con el cursor, solo debemos hacer un clic sobre ella.

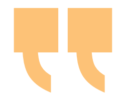
LA VENTANA EXPANDIDA DEL GRÁFICO APARECERÁ COMO UNA VENTANA INDEPENDIENTE



Cursor de referencia

Es posible colocar un cursor de referencia para tomar mediciones de tiempos o amplitudes en los gráficos. Solo debemos presionar la tecla **CTRL** y hacer un clic en el lugar deseado del gráfico. Aparecerá un cursor de color rojo, de referencia, que se puede desplazar o cambiar de señal de forma idéntica al cursor principal mientras mantengamos presionada la tecla **CTRL**; al soltarla, quedará fijo. Ahora podemos desplazar el cursor principal, y en la barra de estado del gráfico, del lado izquierdo, veremos el tiempo en el que se encuentra cada cursor y la distancia entre ambos marcada como **DX**. Del lado derecho, se mostrarán las amplitudes en cada

LOS PUNTOS DE LOS CURSORES PUEDEN ESTAR MIDIENDO SEÑALES DIFERENTES



punto de las señales para los dos cursores, y la diferencia entre ellos marcada como **DY**.

De esta manera, usaremos el cursor de referencia junto con el principal para tomar mediciones, por ejemplo, para calcular la amplitud máxima de una señal o su período.

Debemos tomar en cuenta que los puntos de los cursores pueden estar midiendo señales diferentes, por ejemplo, para medir la diferencia de amplitud entre ambas.

Otras formas de agregar señales a un gráfico

Existen otras formas de agregar señales a un gráfico:

- Al tener un gráfico expandido, podemos usar el botón **Add traces**.
- Ir al menú **Graph** y elegir la opción **Add trace...**.
- También es posible seleccionar uno o más generadores o sondas en el circuito y luego ir al menú **Graph** y elegir la opción **Add trace...**. Al hacer esto, se abrirá una ventana que pregunta: **Quick-add tagged probes?**, que significa: **¿desea agregar rápidamente las sondas y generadores seleccionados?** Si pulsamos **OK**, todas las sondas y generadores seleccionados se agregarán al gráfico en el eje izquierdo y se ordenarán alfabéticamente.

Si elegimos **Add trace...** del menú **Graph** sin haber seleccionado ninguna sonda o generador, o si en la ventana **Quick-add tagged probes?** presionamos el botón **Cancel**, aparecerá una ventana con el título **Add Transient Trace**, donde elegiremos la nueva señal para agregar.



EL BOTÓN SPICE OPTIONS

En las propiedades de los gráficos de simulación aparece un botón llamado **SPICE Options**, que sirve para ajustar las opciones internas del simulador. En algunos casos se pueden ajustar para obtener mejores resultados en el gráfico o para corregir errores, si se presentan. En el **Capítulo 9** hablaremos de este tema.



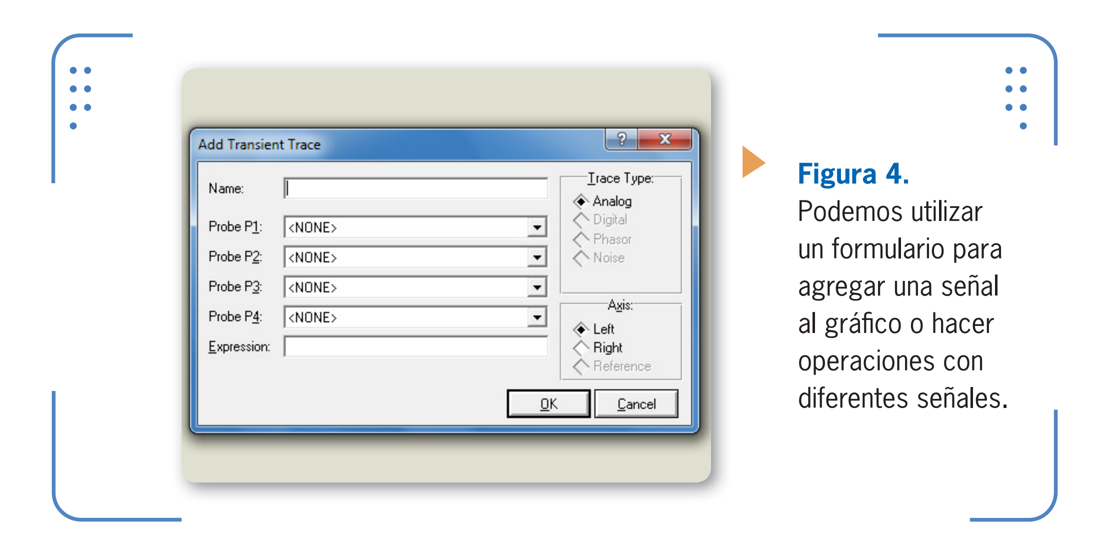


Figura 4. Podemos utilizar un formulario para agregar una señal al gráfico o hacer operaciones con diferentes señales.

En la ventana para agregar una nueva señal, en el campo **Probe P1** debemos especificar el generador o sonda que vamos a utilizar; en la sección **Trace Type** indicamos el tipo de señal (en el caso de este gráfico solo está disponible el analógico), y en la sección **Axis** elegimos el eje vertical izquierdo (**Left**) o derecho (**Right**). Al seleccionar alguna sonda o generador en **Probe P1**, el campo **Name** automáticamente tomará el mismo nombre, aunque si lo deseamos, podemos cambiarlo. Al presionar el botón **OK**, la nueva señal será agregada al gráfico.

Operaciones con las señales

En la ventana **Add Transient Trace**, veremos que hay espacios para elegir más de una sonda o generador en los campos **Probe P1** a **Probe P4**, pero esto solo agregará una señal. Tener más de una señal permite realizar otro tipo de análisis, generando operaciones entre las señales elegidas en los diferentes campos, por ejemplo, sumando dos o más.

Veamos un ejemplo para entender cómo se utiliza esta función. Para esto, descargamos el archivo **Potencia.dsn**, en el que hemos colocado un simple resistor, un generador senoidal y una sonda de corriente. Insertamos un gráfico analógico, y agregamos el generador en el eje izquierdo y la sonda de corriente en el eje derecho. En el menú **Graph**, elegimos la opción **Add traces**, sin tener seleccionado ningún elemento para abrir el cuadro **Add Transient Trace**. Elegimos el generador (**Entrada**) en **Probe P1** y la sonda de corriente en **Probe P2**; notemos que en el

campo **Expression**, aparece **P1+P2**. Como queremos calcular potencia, sabemos que **W=VI**, así que cambiaremos el signo **+** por un * (asterisco), que indica multiplicación. De esta forma se calculará la multiplicación del voltaje del generador por la corriente.

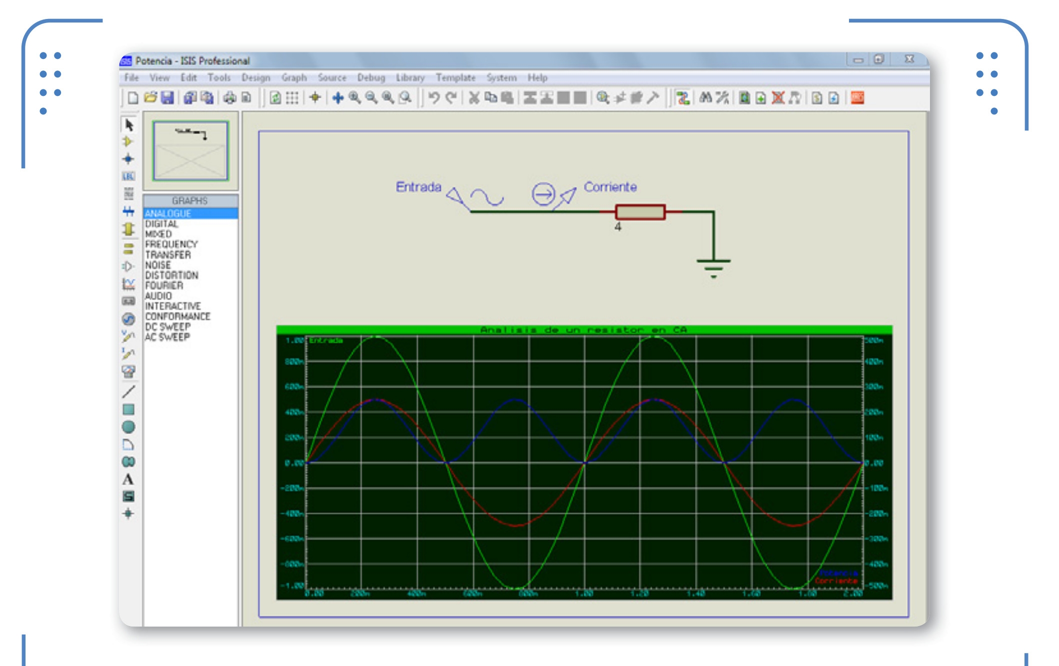


Figura 5. Multiplicando las señales logramos obtener un gráfico de potencia instantánea (representado por la línea azul).

Si seleccionamos el eje derecho (**Right**) en **Axis**, podemos cambiar el nombre de esta señal a **Potencia**, pulsar **OK** y, luego, presionar la **Barra espaciadora** para calcular el gráfico. De esta manera, tendremos un gráfico que nos mostrará el voltaje de la señal, la corriente y la potencia instantánea en el resistor.

Manejar señales y gráficos múltiples

Es posible manejar las señales colocadas en el gráfico. Al hacer un clic en alguna de ellas, quedará seleccionada y podremos arrastrarla para cambiarla de lugar en la lista de señales o ubicarla en el eje vertical contrario. Si hacemos un clic derecho sobre la etiqueta de

alguna señal, en el menú contextual encontraremos las opciones **Drag Trace Label** (arrastrar etiqueta de señal) para moverla, **Edit Trace Properties** (editar propiedades) para acceder a sus propiedades y cambiar su nombre o la operación entre las señales, y **Delete Trace** para eliminar la señal del gráfico.

Podemos colocar más de un gráfico en nuestro diseño, pero solo se calculará uno a la vez. Si tenemos más de uno, debemos ubicar el cursor del mouse sobre el gráfico que queremos calcular y presionar la **Barra espaciadora**. En el menú **Graph**, se mostrará la lista de gráficos con sus nombres, y el gráfico seleccionado o el último gráfico calculado tendrán una marca. Al elegir alguna de las opciones de este menú, por ejemplo **Simulate Graph**, actuará sobre el gráfico marcado en la lista.

Cálculo de transformadores

En algunos proyectos será necesario simular transformadores, por eso debemos saber cómo calcularlos. Veamos el procedimiento para realizar el cálculo de transformadores simples y con derivación central. Como sabemos, la relación de transformación está dada por:

$$n = \frac{Ns}{Np} = \frac{Vs}{Vp} = \sqrt{\frac{Ls}{Lp}}$$



IDENTIFICAR PRIMARIO Y SECUNDARIO

En los transformadores simples es importante no tomar los bobinados de forma incorrecta. Antes de colocar el transformador en el diseño, debemos asegurarnos de que el ángulo en la barra de rotación y reflexión sea de **cero grados**; así, el lado izquierdo será el primario, y el lado derecho, el secundario. Si el ángulo es de **180 grados**, los bobinados estarán invertidos.

Donde:
n = Relación de transformación
Ns = Número de espiras del secundario
Np = Número de espiras del primario
Vs = Voltaje en el secundario
Vp = Voltaje en el primario
Ls = Inductancia del secundario
Lp = Inductancia del primario

En Proteus, es posible configurar las propiedades de los transformadores modificando la inductancia de los bobinados y el factor de acoplamiento. Para calcular el valor de un transformador podemos despejar el valor de **Ls**:

$$Ls = \left(\frac{Vs}{Vp}\right)^2 Lp$$

De esta manera, es posible calcular la inductancia del secundario dando una inductancia del primario, con los voltajes de primario y secundario deseados, y, además, tomando en cuenta un factor de acoplamiento que sea igual a 1.

Si descargamos y abrimos el archivo **TransformadorSimple.dsn**, podremos acceder a un muy buen ejemplo del cálculo de transformadores simples en Proteus.



TIERRA EN TRANSFORMADORES

En las simulaciones, para el cálculo de transformadores se ha colocado tierra en ambos bobinados. Esto se debe a que Proteus necesita siempre de una referencia para calcular las simulaciones, y si no la hay, pueden producirse errores u obtenerse resultados equivocados en las mediciones.

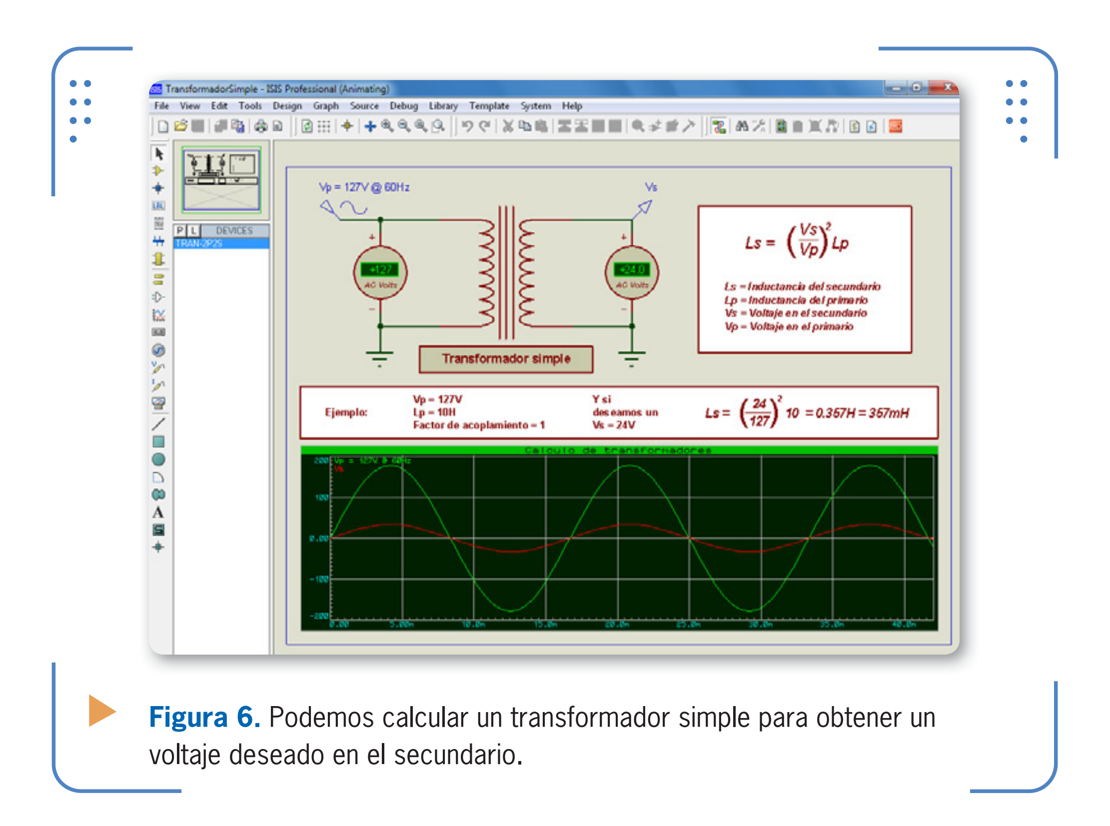


Figura 6. Podemos calcular un transformador simple para obtener un voltaje deseado en el secundario.

Y para el transformador con derivación central, el cálculo se hará con la ecuación que expresamos a continuación:

$$Ls = \frac{\left(\frac{Vs}{Vp}\right)^2 Lp}{2}$$

Para observar este caso, tenemos un ejemplo en el archivo **TransformadorDeriv.dsn**, donde es posible calcular transformadores de reducción o elevación para las simulaciones de los circuitos.

Solo tendremos que descargar y abrir el archivo, y observar con atención el funcionamiento.

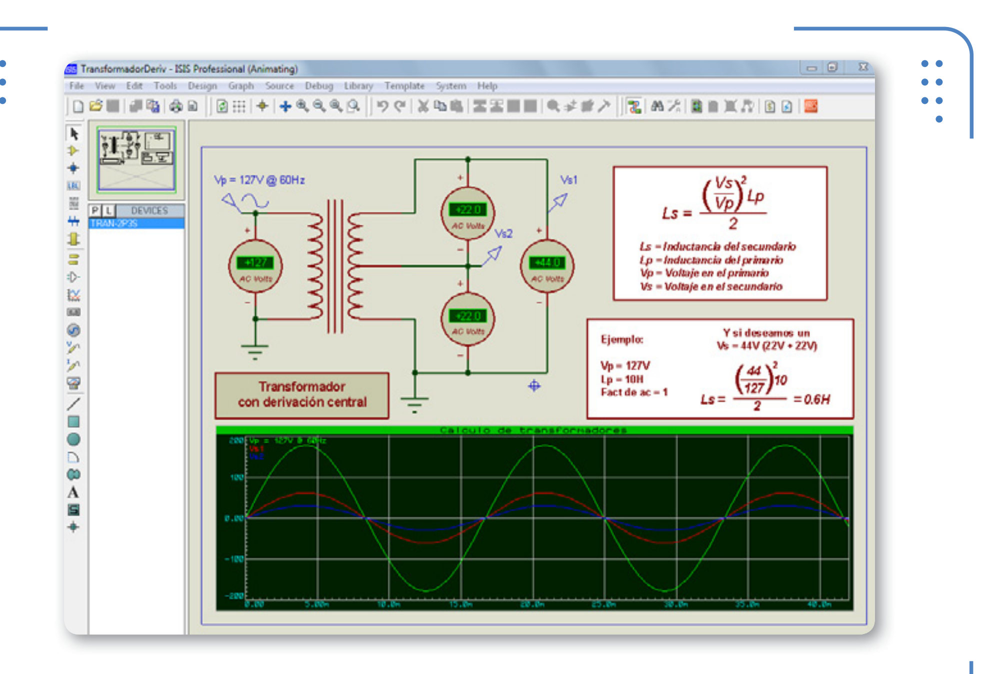


Figura 7. Los transformadores con derivación central también pueden ser calculados para un valor específico a la salida.



Gráfico digital

El siguiente gráfico es el digital y, tal como su nombre lo indica, analizará señales en circuitos digitales; en esencia, es el equivalente al analizador lógico. Al igual que en el gráfico analógico, basta con colocar un gráfico digital en la ventana de ISIS, agregar las señales mediante generadores o sondas y calcular el gráfico para tener un análisis de las señales digitales.

Para los gráficos digitales solo deben usarse generadores digitales y sondas de voltaje. Si usamos sondas de corriente, ISIS se forzará a usar simulación analógica en vez de digital, con la correspondiente carga que esto implica, así que no tiene sentido hacerlo. Como ejemplo podemos ver el archivo **GraficoDigital.dsn**, en el que hemos colocado el mismo circuito integrado 4022 usado con el analizador lógico, pero esta vez, con un gráfico digital.

Figura 8. También se puede analizar el funcionamiento de un circuito integrado 4022 con un gráfico digital.

En este ejemplo, colocamos las sondas en las salidas del 4022 y un bus para ver cómo se presenta en el gráfico digital (violeta). El gráfico digital también puede ser expandido y analizado con el cursor; los estados se mostrarán al lado del nombre de cada señal. Solo permite el uso del eje vertical izquierdo.



Gráfico mixto

Si tenemos un circuito mixto, podemos usar gráficos analógicos y digitales en el mismo circuito para analizarlo. Sin embargo, puede ser útil tener una visualización de las señales en un solo gráfico, y para esto existe un gráfico mixto llamado **MIXED**. Inicialmente, funciona de forma idéntica al analógico al colocar el gráfico mixto; si este se encuentra vacío y le agregamos cualquier generador o sonda, estas se incorporarán como señales analógicas. Para agregar la primera señal

digital, debemos usar la opción **Add trace...** y, en la ventana **Add Transient Trace**, seleccionar **Digital** en la sección **Trace Type**. Una vez que tenemos al menos una señal digital, podemos arrastrar cualquier generador hacia el gráfico y soltarlo en la sección digital (la parte superior) o analógica (parte inferior). Tenemos un ejemplo de un gráfico mixto en el archivo **GeneradorEscalera.dsn**. Solo debemos descargarlo, abrir el archivo y prestar atención al procedimiento.

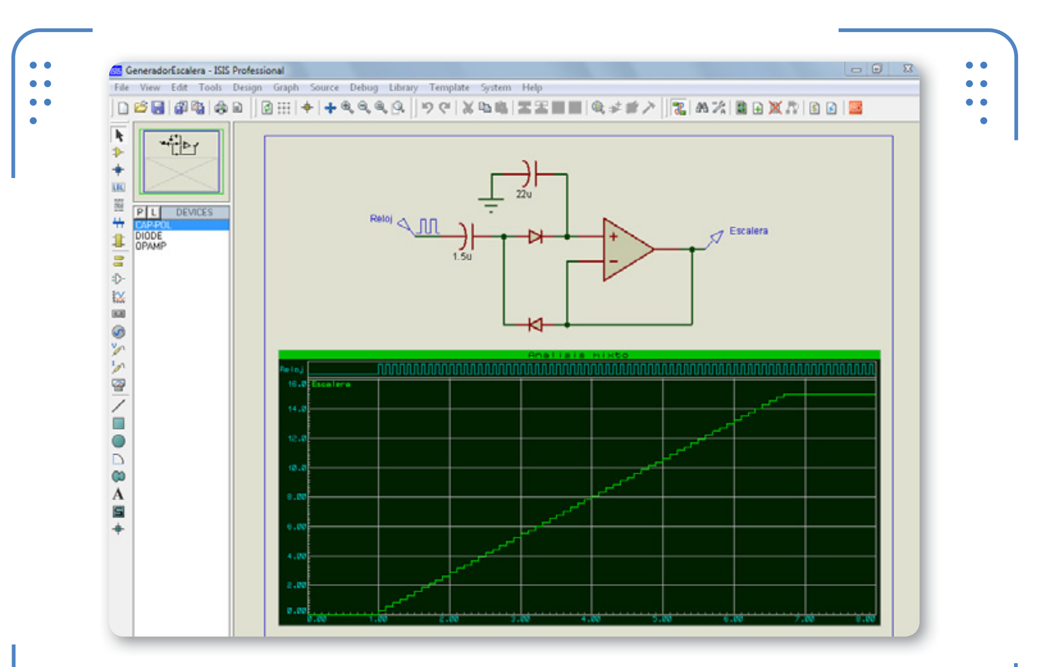


Figura 9. Un generador de escalera analógica con una señal digital de reloj es buen un ejemplo del análisis mixto.

AUDIO CONTRA TIEMPO DE CÁLCULO

Cuando usamos el gráfico de audio para analizar algunos circuitos que manejen o generen sonido, debemos elegir, en la medida de nuestras posibilidades, tiempos cortos para el eje horizontal del gráfico. En el caso de que elijamos tiempos demasiado largos, el cálculo del gráfico puede llegar a demorar mucho tiempo, dependiendo de cuál sea la carga de la CPU que se genere. Por esta razón, siempre debemos pensar en las variables que influirán en los tiempos de simulación.

Este circuito es un generador de escalera que usa una señal digital de reloj para producir una escalera analógica usando un amplificador operacional. Notemos como detalle en este circuito que hemos utilizado solo componentes genéricos.



Gráfico de audio

El gráfico de audio nos permite hacer análisis de los circuitos que generan sonido. En principio, es un gráfico analógico, pero con la diferencia de que, al graficar la señal, la convertirá en un archivo tipo **WAV** y la reproducirá a través de los altavoces de la computadora, lo cual permite escuchar los sonidos generados en la simulación. Para ejemplificar el uso del gráfico de audio, utilizaremos un circuito que genera un sonido de alarma, construido con un par de circuitos integrados 555 o un 556. Debemos descargar el archivo **Sirena.dsn**.

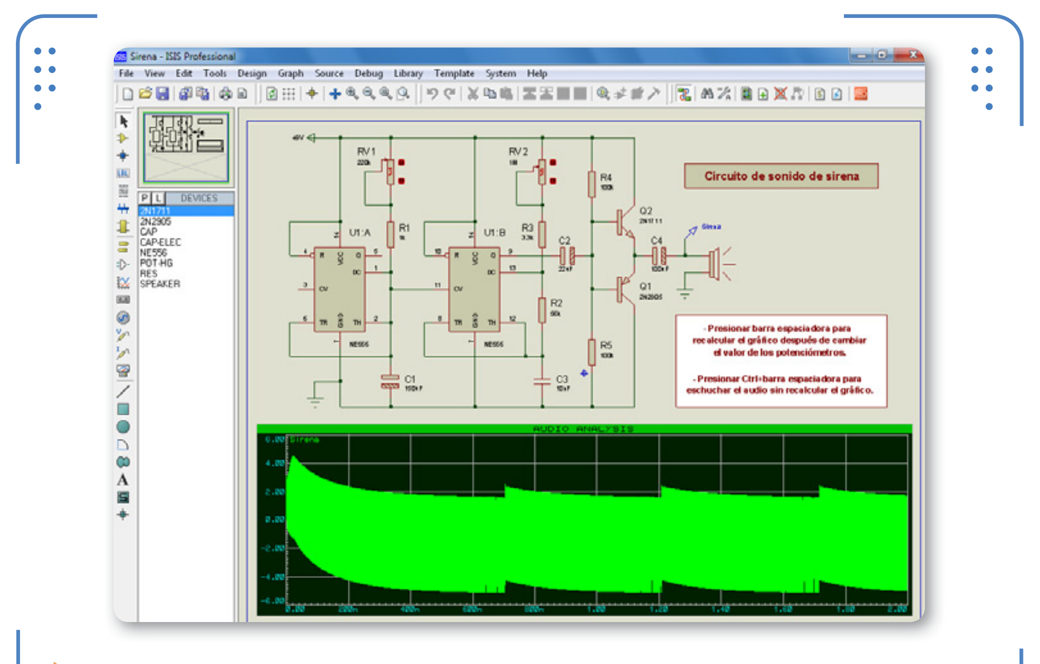


Figura 10. Si tenemos circuitos que generen sonido, es posible analizarlos con el gráfico **AUDIO**.

En este circuito notaremos que, si corremos la simulación de forma interactiva (con el botón **Play**), la carga de la CPU puede ser alta, por lo que no correrá en tiempo real y el sonido no será adecuado para escuchar el efecto. Si usamos el gráfico de audio, podremos escuchar el sonido de manera correcta.

Las propiedades del gráfico de audio son iguales a las del analógico, excepto por algunas funciones de reproducción para el audio generado. Al igual que en el gráfico analógico, en el de audio se puede agregar más de una señal o usar ambos ejes verticales. En la siguiente **Guía visual** conoceremos en detalle las propiedades.

GUÍA VISUAL ■ PROPIEDADES DE AUDIO

01 **LOCK TIMESTEP (FIJAR TIEMPO DE PASO):** esta opción fija el tiempo de paso interno del simulador para que concuerde con el elegido en la frecuencia de muestreo. Puede ser útil para acelerar el tiempo de cálculo del gráfico si estamos calculando tiempos largos, aunque puede ser menos preciso.

02 **PLAY AUDIO (REPRODUCIR AUDIO):** permite reproducir el audio del gráfico sin calcularlo de nuevo.

03 **EXPORT AS WAV FILE (EXPORTAR ARCHIVO WAV):** para guardar la señal de audio como un archivo **WAV**.

04 **SAMPLE RATE (FRECUENCIA DE MUESTREO):** aquí elegimos la frecuencia de muestreo para la señal, entre 11025, 22050 o 44100 KHz.

05 **SAMPLE BITS (BITS DE MUESTREO):** aquí especificamos el número de bit de muestreo para la señal de audio, 8 o 16 bits.

06 **LOOP TIME (TIEMPO DE REPETICIÓN):** este campo define el tiempo de repetición de la señal de audio, en segundos. Esto significa que la señal generada se repetirá en este intervalo; es útil para generar archivos **WAV** más largos sin tener que simular tiempos muy extensos.

07 **PLAY MODE (MODO DE REPRODUCCIÓN):** aquí elegimos el modo de reproducción entre las siguientes opciones:

- **MONO:** se reproducirá la suma (mezcla) de todas las señales del gráfico de forma monoaural.
- **STEREO:** se reproducirá la señal del eje izquierdo en el canal izquierdo, y la del eje derecho en el canal derecho de forma estéreo.
- **INPUT:** se reproducirá la suma de todas las señales en el eje izquierdo de forma monoaural.
- **OUTPUT:** se reproducirá la suma de todas las señales en el eje derecho de forma monoaural.



CURSORES Y FLECHAS DEL TECLADO

Podemos usar el teclado de nuestra computadora para desplazar los cursores en un gráfico ampliado. Las flechas **derecha** e **izquierda** se usan para desplazar el cursor en sentido horizontal, y las flechas de **arriba** y **abajo** se utilizan para cambiar de señal. En el caso del cursor de referencia será necesario presionar también la tecla **CTRL** y luego manipularlo con el teclado. Recordemos que, además, en muchos de los menús, al costado de las opciones, tenemos las teclas que podemos usar como atajos.



Gráfico de respuesta en frecuencia

El siguiente gráfico que veremos es el de respuesta en frecuencia, que en la lista de gráficos aparece como **FREQUENCY**. Este análisis se realiza en el dominio de la frecuencia contra la amplitud de una señal y también la fase. Es útil para graficar la respuesta en frecuencia en circuitos como amplificadores y filtros. También es conocido como **diagrama de Bode**.

Amplificador de audio

Veamos un ejemplo práctico del uso del gráfico de respuesta en frecuencia analizando un amplificador construido con un TDA2030. Tenemos que descargar y abrir el archivo **TDA2030.dsn**. Para realizar el análisis en frecuencia, debemos agregar un gráfico **FREQUENCY** y la salida del amplificador mediante la sonda **Audio out** en ambos ejes verticales, de modo de graficar tanto la amplitud como la fase. El eje izquierdo indicará **GAIN (dB)**, y el derecho dirá **PHASE**.

EL GRÁFICO FREQUENCY SE UTILIZA PARA ANALIZAR FILTROS Y AMPLIFICADORES



Además, tenemos que elegir una señal de referencia para el análisis, que es el generador **Ref**. Para generar el gráfico, Proteus inyecta una señal senoidal de frecuencia variable de manera automática, pero necesita saber en qué punto inyectarla. Es por eso que se coloca el generador senoidal a la entrada del amplificador, sin importar su configuración. Es posible arrastrar



GRÁFICOS CON SEÑALES COMPLEJAS

Es importante tomar en cuenta que si tenemos un gráfico con tiempos muy extensos o señales complejas, al guardar el diseño como un archivo **.DSN**, la señal dibujada en el gráfico se guardará en él y puede generarse un archivo muy pesado. También el módulo ISIS puede congelarse o colapsar por falta de memoria en el sistema al intentar calcular un gráfico muy complejo.

el generador de referencia al gráfico, y este no se agregará como una señal, sino que será tomado como la referencia, o se puede elegir en las propiedades del gráfico el campo **Reference**.
Además, en las propiedades debemos configurar la frecuencia inicial y final para el análisis, los intervalos (décadas, octavas o lineal) y el número de pasos por intervalo (**Steps/Interval**). Cuanto mayor sea este número, más detallado será el gráfico, pero más tiempo tomará calcularlo. Podemos elegir también una escala lineal para el eje vertical.

En el ejemplo, hemos colocado dos generadores, uno de audio para analizar la señal de audio en la entrada y la salida (el gráfico de abajo), y uno senoidal para hacer el análisis de respuesta en frecuencia (derecha). Si intentamos hacer el análisis de frecuencia con el generador de audio, obtendremos un error, así que es importante desconectar siempre un generador, dependiendo de qué gráfico queremos calcular.

> CUANTO MAYOR SEA EL NÚMERO DE PASOS POR INTERVALO, MÁS DETALLADO SERÁ EL GRÁFICO



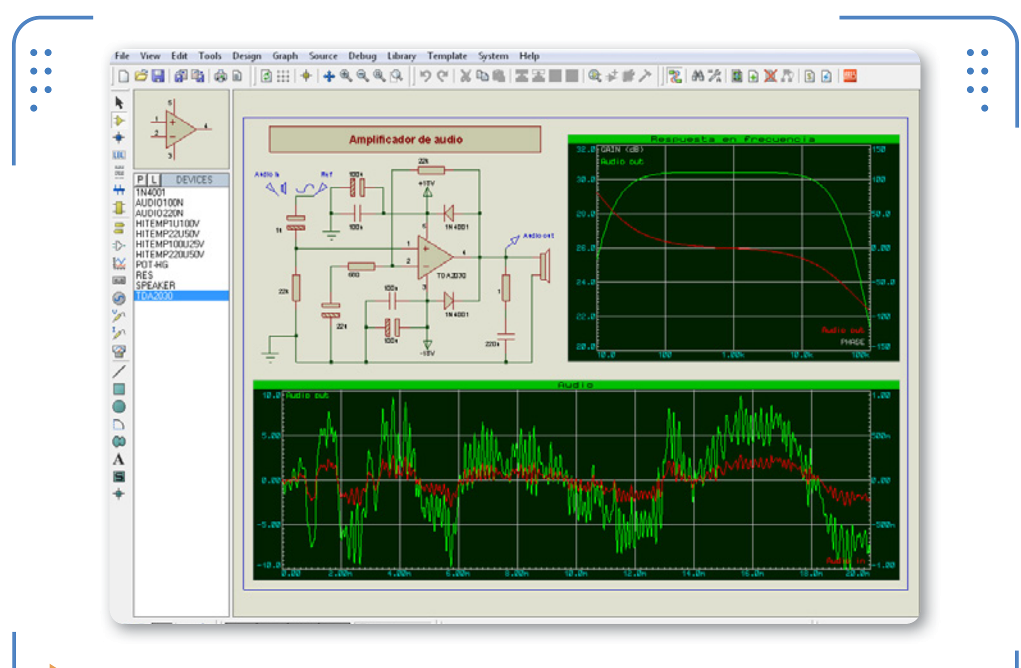


Figura 11. En este ejemplo realizamos el análisis de la respuesta en frecuencia de un amplificador de 14 Watts.

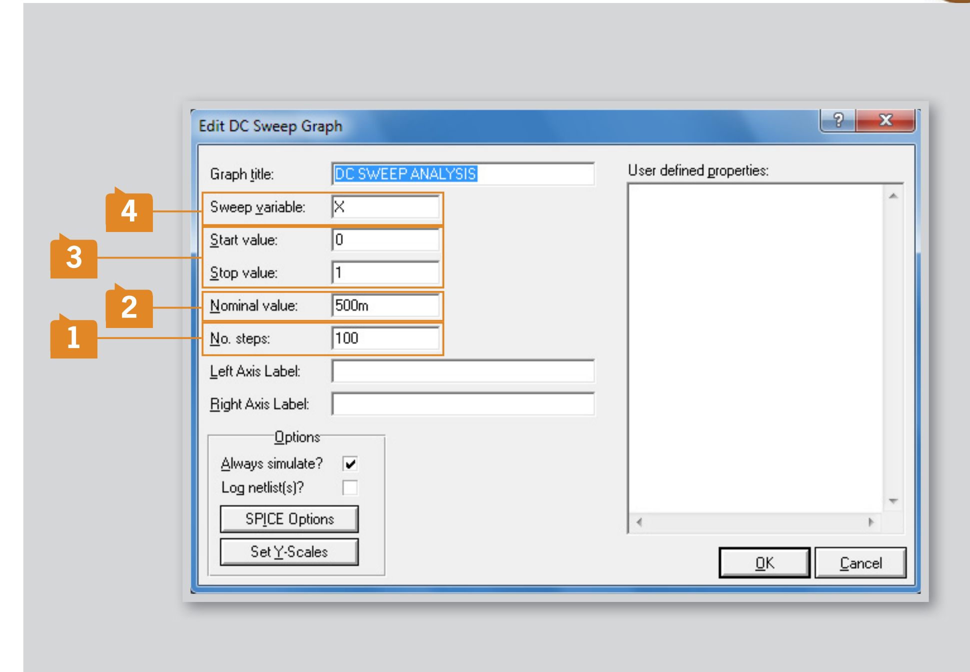


Gráfico de barrido en CD

El siguiente gráfico que nos será de utilidad es el de barrido en CD, denominado **DC SWEEP**. En él vamos a definir una variable para generar un gráfico del comportamiento del circuito al cambiar dicha variable, que puede ser voltaje, resistencia, capacitancia, etcétera. En la siguiente **Guía visual** conoceremos sus propiedades.

▼ GUÍA VISUAL ■ PROPIEDADES DEL DC SWEEP







NO. STEPS (NÚMERO DE PASOS): permite definir cuántos pasos se harán al calcular el gráfico; a mayor valor, más detallado será el gráfico. Si obtenemos un gráfico discontinuo o con apariencia angulosa, podemos aumentar este valor.



NOMINAL VALUE (VALOR NOMINAL): aquí establecemos el valor nominal de la variable. Esto ayudará al simulador a encontrar el punto de operación del circuito al iniciar el análisis. Típicamente, debe ser un valor que se encuentre dentro del rango de valores de inicio y fin. ▶



START VALUE/STOP VALUE (TIEMPO DE INICIO Y FIN): estos campos definen los valores inicial y final del gráfico. Debemos tomar en cuenta que dependen de la variable elegida. Si elegimos capacitancia, tenemos que asignar valores de esta clase, por ejemplo, 1u/1000u, para variar la capacitancia de 1 a 1000 microfaradios.



SWEEP VARIABLE (VARIABLE DE BARRIDO): para indicar el nombre de la variable que vamos a usar, puede ser una letra (X, Y, R C… etc.) o una expresión, por ejemplo Res, Temp, etc. También podemos ingresar un nombre descriptivo, por ejemplo, C o Cap, para indicar que nuestra variable será capacitancia.

Después de elegir el nombre de la variable en las propiedades del gráfico, debemos asignar ese mismo nombre al componente adecuado. Por ejemplo, si vamos a variar el valor de un capacitor, en las propiedades debemos asignar el nombre de dicha variable como su valor. Veamos un ejemplo para entender bien cómo funciona este análisis; usaremos el archivo **DCsweep.dsn**, en el que tenemos el análisis de un resistor.

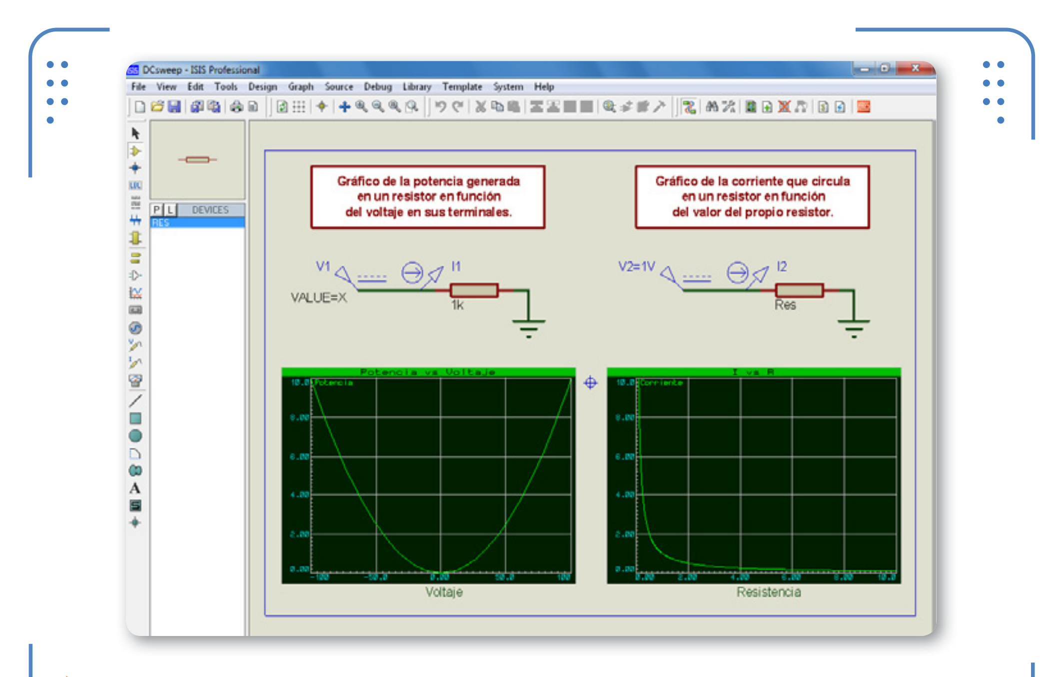


Figura 12. Ejemplos del uso del gráfico de barrido en CD al analizar el comportamiento de un simple resistor.

En el primer gráfico se analiza la potencia generada en el resistor al variar el voltaje de entrada, asignando la variable **X** al generador de DC. Para esto, debemos editar sus propiedades como texto, colocar **VALUE=X**, y elegir los valores de inicio y fin de **-100** a **100**, que representan voltaje. De esta forma, el eje horizontal muestra el voltaje, mientras que el eje veertical presenta la potencia, al haber agregado la señal **P1*P2**, que es la multiplicación del voltaje del generador por la corriente en la sonda.

En el segundo gráfico obtenemos un análisis de cómo cambia la corriente en el resistor al variar su resistencia. Esta vez la variable es **Res**, que representa la resistencia; en las propiedades del resistor hemos colocado **Res** como valor y agregamos la sonda de corriente al gráfico. Así vemos cómo se hace un análisis del comportamiento del circuito al variar un parámetro en él usando el gráfico de barrido en CD.



Gráfico de barrido en CA

El siguiente gráfico es el de barrido en CA, cuyo nombre es **AC SWEEP**. Para definirlo rápidamente podemos decir que es un gráfico de respuesta en frecuencia múltiple. Es útil para observar cómo un parámetro afecta la repuesta en frecuencia de un circuito. Sus propiedades son una combinación entre las del gráfico de respuesta en frecuencia y las del gráfico de barrido de CD. Debemos especificar un generador de referencia y una variable.

Recordemos el circuito del amplificador inversor que usamos antes, lo tomaremos como ejemplo. Como sabemos, la ganancia de este amplificador está dada por la relación entre los resistores **RIN** y **RF**, así que podemos ver cómo afecta dicha ganancia el cambio de uno de ellos, por ejemplo, **RF**.



NO HAY BARRIDOS NEGATIVOS

En el gráfico de barrido en CD debemos especificar un valor inicial y otro final para el barrido de la variable. El inicial puede ser negativo, pero el final debe ser siempre mayor a este, ya que no es posible realizar barridos negativos; la variable debe aumentar, nunca disminuir.

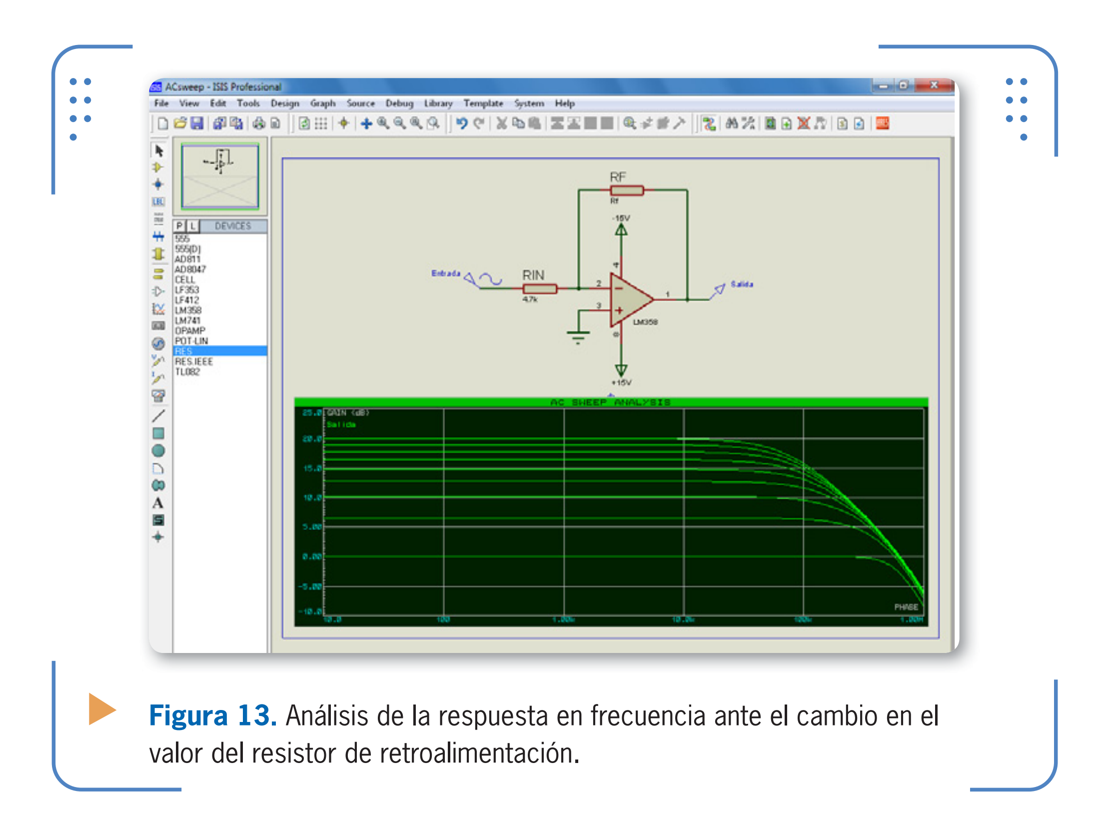


Figura 13. Análisis de la respuesta en frecuencia ante el cambio en el valor del resistor de retroalimentación.

Para este análisis hemos definido la variable como **Rf**, con un valor que variará desde 4.7 k a 47 k, de esta manera podremos ver la respuesta en frecuencia del amplificador ante el cambio de valor. Además, elegimos 10 pasos, es decir, 10 diferentes valores para generar el gráfico. El módulo ISIS analiza el valor máximo y mínimo de la variable y, luego, la divide entre 10, en este caso para generar 10 gráficos. De esta forma, podremos observar cómo la ganancia del amplificador crece mientras aumenta el valor de **RF**.

HASTA 20 PASOS EN AC SWEEP

En el gráfico de barrido en CA es posible elegir el número de pasos (**No. Steps**), que es la cantidad de gráficos que veremos en el análisis. Este número debe ir de 1 hasta 20 como valores mínimo y máximo; porque si colocamos un número mayor que 20 o menor que 1, aparecerá un aviso que nos advierte que el valor elegido no está dentro del rango permitido.



Gráfico interactivo

Al comenzar el capítulo, explicamos que el análisis con gráficos es no interactivo, pero el denominado **INTERACTIVE** es la excepción. Se trata de un **gráfico mixto**, donde podemos analizar señales analógicas y digitales, con la diferencia de que, al calcular el gráfico, la simulación interactiva correrá también mientras se grafica la señal en él. Esto resulta útil en aquellos casos en que necesitamos monitorear el efecto de algún elemento interactivo, por ejemplo, un pulsador o un potenciómetro en el circuito, o para hacer un análisis de un circuito en el que hay elementos puramente interactivos, como el generador de señales o la terminal virtual. Precisamente, tenemos un sencillo ejemplo del gráfico interactivo en la simulación de la comunicación entre terminales virtuales que estudiamos en el **Capítulo 4**. En el archivo **GraficoInteractivo.dns**, vemos esta simulación con algunas modificaciones: la terminal transmisor tiene un texto automático que se enviará al inicio de la simulación y se ha agregado el gráfico interactivo.

INTERACTIVE ES UN GRÁFICO MIXTO PARA ANALIZAR SEÑALES ANALÓGICAS Y DIGITALES



Al calcular el gráfico presionando la **Barra espaciadora**, la simulación correrá de 0 a 2 segundos, ya que así se configuró el tiempo en el gráfico interactivo. De esta forma, todo lo que ocurra en las sondas durante ese tiempo será graficado y mostrado al terminar la simulación, la cual se detendrá automáticamente, y en el gráfico aparecerán las señales del envío de datos entre las terminales virtuales. En el ejemplo, hemos graficado tanto en digital como en analógico para mostrar ambos, y agregamos dos sondas ya que no es posible graficar los dos modos con una sola.

NÚMERO DE PASOS EN AC SWEEP

En el análisis de barrido de CA se pueden colocar hasta 20 pasos, pero si elegimos un número grande, el cálculo se torna complejo, sobre todo, si se usa ganancia y fase. En este caso, ISIS puede congelarse o colapsar, por lo que es recomendable no usar más de diez pasos a menos que sea necesario.

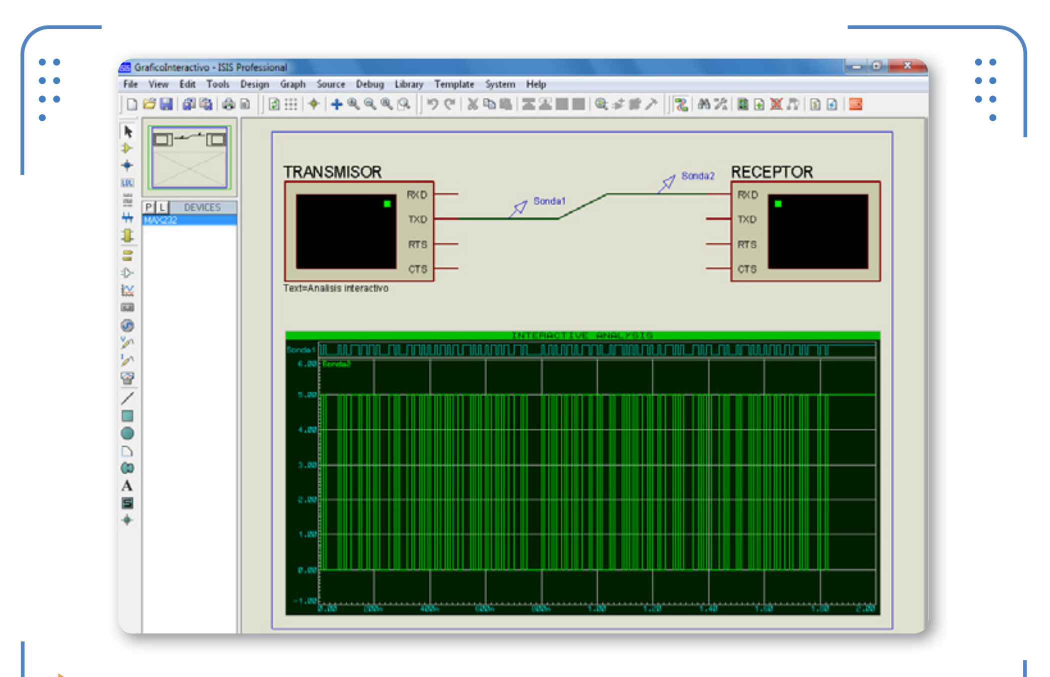


Figura 14. El gráfico interactivo nos permite analizar la transmisión entre las terminales virtuales.



Gráfico de transferencia en CD

El gráfico de transferencia en CD, que en el listado de ISIS recibe el nombre **TRANSFER**, está diseñado para modelar las curvas características de los dispositivos semiconductores.

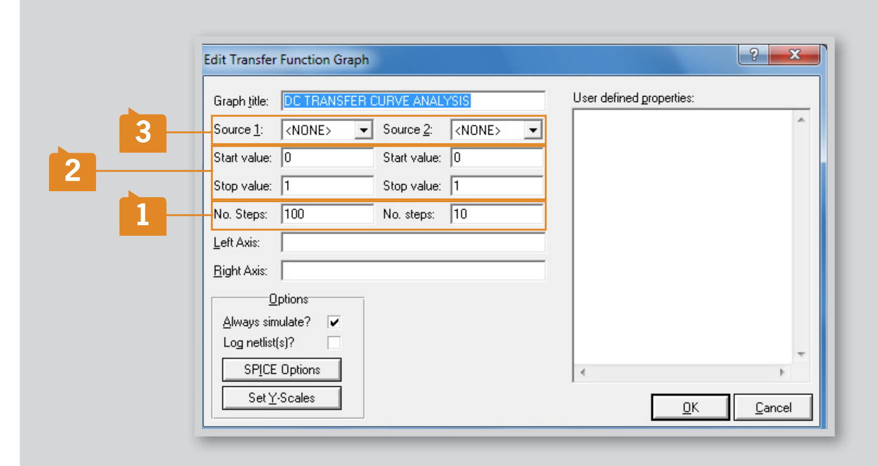
Realmente podemos verlo como un gráfico de barrido en CD múltiple, ya que en él se puede hacer un barrido con dos valores; aunque no se usan variables, los valores de barrido siempre serán fuentes de voltaje o corriente. En la siguiente **Guía visual** conoceremos sus propiedades en detalle.

EN TRANSFER, LOS VALORES DE BARRIDO SERÁN FUENTES DE VOLTAJE O CORRIENTE



▼ GUÍA VISUAL ■ PROPIEDADES DEL TRANSFER





01 **NO. STEPS (NÚMERO DE PASOS):** aquí se indica cuántos pasos se usarán para cada fuente; la fuente 1 tiene un límite de 10000, mientras que la fuente 2, solo de 20 pasos.

02 **START VALUE/STOP VALUE (TIEMPO DE INICIO Y FIN):** estos campos definen el valor inicial y final de cada fuente, respectivamente; se eligen dependiendo del tipo de fuente y los rangos deseados.

03 **SOURCE 1/SOURCE 2 (FUENTE 1/FUENTE 2):** en estos campos se determinan las fuentes de voltaje o corriente para el análisis; se puede elegir solo una o ambas, dependiendo del análisis deseado.

Como ejemplo, veamos las curvas características de un transistor en el archivo **Curvas2N3904.dsn**. Para calcular este gráfico, lo haremos



TRANSFERENCIA DE DIODOS

El gráfico TRANSFER está diseñado para trazar curvas características de dispositivos semiconductores, aunque se puede usar en otras tareas. Como ejemplo, descarguemos el archivo **TranferenciaDiodos.dsn** donde graficamos el comportamiento de un diodo rectificador 1N4001, y de un zener 1N4370A.

de la siguiente forma: elegimos un transistor, en este caso, el 2N3904; en la base colocamos una fuente de corriente llamada **Ib**, que proporcionará una corriente de base variable; en el colector agregamos una fuente de voltaje llamada **Vce**, que proporcionará un voltaje variable al colector; y tenemos una sonda de corriente **Ic** que medirá la corriente en el colector.

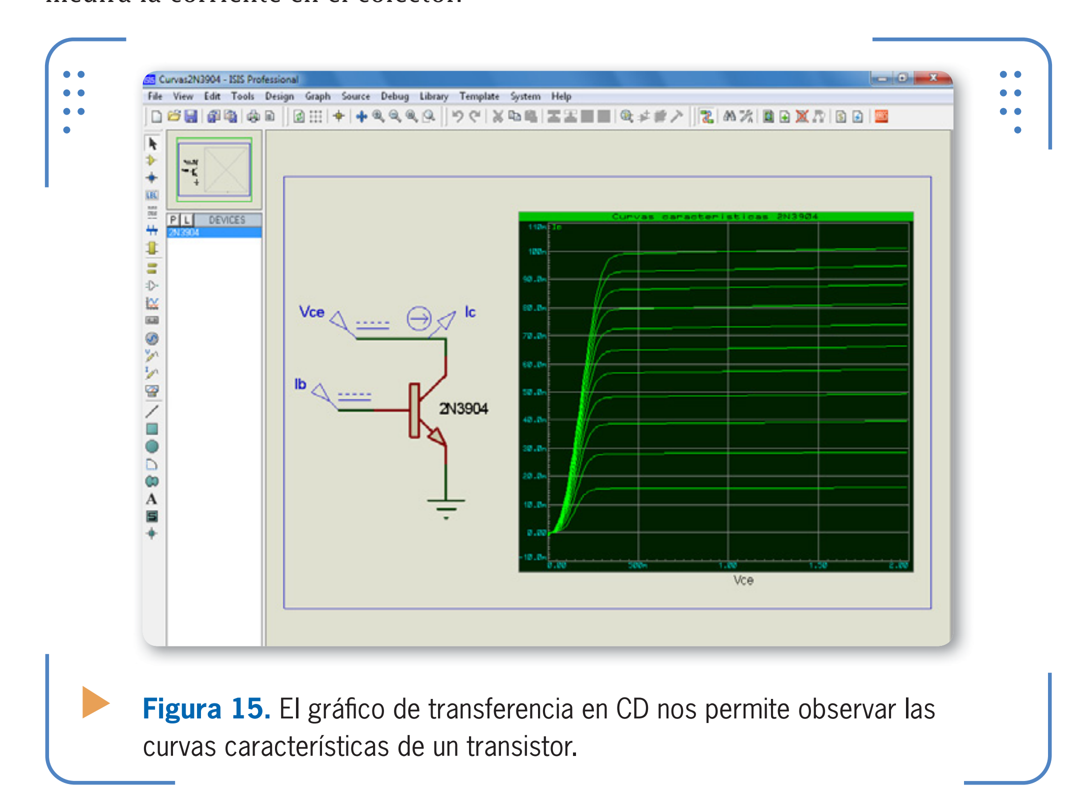


Figura 15. El gráfico de transferencia en CD nos permite observar las curvas características de un transistor.

Notemos que las fuentes no necesitan configurarse, porque al ser un gráfico de transferencia serán variables de forma automática. Ahora agregamos el gráfico TRANSFER y en la propiedades configuramos la fuente 1 (**Source 1**) como **Vce**, con un rango de 0 a 2 Volts, y un número de pasos de 200. En la fuente 2 (**Source 2**) elegimos **Ib** con un rango de 100 uA a 1 mA y 10 pasos. Al calcular el gráfico (presionando la **Barra espaciadora**), obtenemos las curvas características del transistor. Cada línea del gráfico representa la corriente de colector IC generada con una corriente de base específica (por ejemplo, 1 mA para la curva más alta) y cómo varía al aumentar el Vce.



Cycles/Graph en los generadores SINE y PULSE

En el **Capítulo 3**, vimos que en los generadores SINE y PULSE existe una opción en sus propiedades llamada **Cycles/Graph** o ciclos por gráfico, que sirve para generar un número determinado de ciclos de señal, el cual dependerá de un gráfico de simulación. Si agregamos un gráfico de tipo analógico, de audio o interactivo, e incorporamos el generador, con esta opción se producirán solo los ciclos definidos a lo largo del gráfico. Es decir, si establecemos los ciclos por gráfico en 10, y el gráfico tiene un tiempo configurado de 0 a 5 segundos, entonces se generarán 10 ciclos de señal en esos 5 segundos.

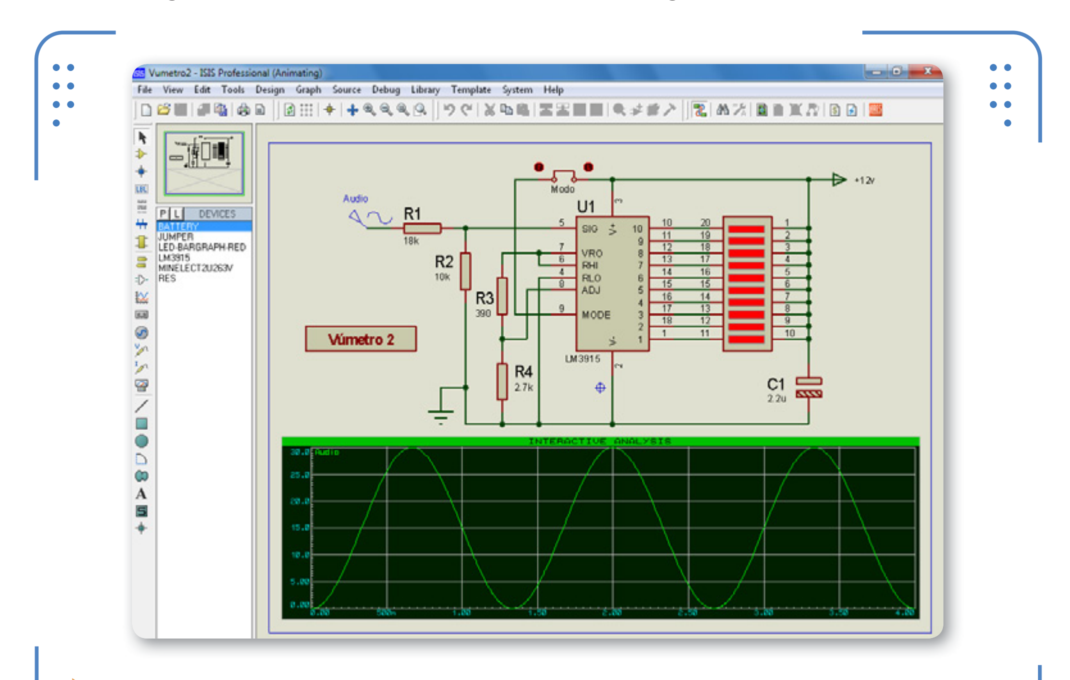


Figura 16. Se puede combinar un gráfico con la opción **Cycles/raph** en los generadores **SINE** y **PULSE**.

Veamos un ejemplo. Tomamos el circuito del vúmetro del **Capítulo 3**, reemplazamos el generador de audio por un **SINE**, agregamos un **gráfico**

interactivo y lo configuramos con un tiempo de 0 a 4 segundos. En el generador seleccionamos la opción **Cycles/Graph** con un valor de 3.

Debemos descargar y abrir el archivo **Vumetro2.dsn**; al presionar la **Barra espaciadora**, podremos ver la simulación corriendo, y al terminar, se dibujará la señal en el gráfico. Siempre serán tres ciclos, sin importar la configuración de tiempo del gráfico.



RESUMEN

Los gráficos de simulación son otra poderosa herramienta de análisis en Proteus. En este capítulo, hemos estudiado cómo podemos agregarlos en nuestros diseños, cómo simularlos, y cómo analizar las señales en los gráficos analógicos y digitales más importantes. Esto nos permite realizar un profundo estudio de las señales y de otros parámetros de los circuitos, para ver si dichas señales corresponden a lo que esperamos del circuito. A partir de ahora, los gráficos de simulación se convertirán en una herramienta de uso cotidiano para nuestro trabajo con Proteus.

Actividades

TEST DE AUTOEVALUACIÓN

1	¿Para qué sirve un gráfico de simulación?
2	¿Cuál es la diferencia entre los gráficos de simulación y los instrumentos virtuales?
3	¿Cómo se coloca un gráfico?
4	¿Cómo se define el punto de prueba de una señal para un gráfico?
5	Describa un método para agregar una señal a un gráfico.
6	¿Cómo se calcula un gráfico?
7	¿Cómo se amplía la ventana de un gráfico?
8	¿Para qué sirve un gráfico mixto?
9	¿Para qué sirve un gráfico de audio?
10	¿Cuál es la diferencia entre un gráfico interactivo y los demás tipos de gráficos?

ACTIVIDADES PRÁCTICAS

1	Abra el archivo **PWM.dsn** utilizado en el **Capítulo 4**, borre el osciloscopio y, en su lugar, analice las señales con un gráfico, que puede ser analógico o mixto.
2	En un nuevo diseño, coloque un circuito integrado 4024, un generador DCLOCK y sondas en las salidas. Visualice su funcionamiento con un gráfico digital.
3	Abra el archivo **TransformadorDeriv.dsn** y calcule Ls para obtener un voltaje de salida de 24 V (12 V + 12 V) en el secundario. Compruebe el resultado corriendo la simulación y en el gráfico.
4	Abra el archivo **Curvas2N3904.dsn** y reemplace el transistor por un 2N3055; calcule de nuevo el gráfico para ver la diferencia entre un transistor de pequeña señal y un transistor de potencia (2N3055).
5	Abra el archivo **Vumetro2.dsn**, cambie el tiempo del gráfico con un valor de 0 a 20 segundos, presione la barra espaciadora y observe qué sucede.



Simulación con microcontroladores PIC

Para quienes diseñan circuitos con microcontroladores, Proteus representa una herramienta muy poderosa porque permite verificar si los programas funcionan correctamente. Además, cuenta con múltiples herramientas de análisis y depuración para localizar errores y corregirlos de forma rápida. En este capítulo, estudiaremos la simulación de microcontroladores PIC.





Microcontroladores en Proteus

Proteus cuenta con una amplia gama de **microcontroladores** de diferentes familias que pueden simularse con todas sus funciones; los principales son: PIC y dsPIC, AVR, BASIC Stamp, 8051 de Intel, MSP430 y PICAXE, entre otros. Además, posee un gran número de **periféricos** que pueden interactuar con los microcontroladores, como: RTCs, convertidores A/D y D/A, memorias, potenciómetros digitales, sensores de temperatura, etcétera. Para acceder a los microcontroladores y periféricos disponibles, en la ventana **Pick Devices** elegimos la categoría **Microprocessor ICs** o también podemos realizar una búsqueda específica.

LOS BUSES CONECTAN MÚLTIPLES LÍNEAS PARA QUE NO HAYA UN EXCESO DE ELLAS

La gran mayoría de los simuladores de microcontroladores solo permiten simular el funcionamiento del propio microcontrolador, pero no es posible observar el comportamiento de elementos externos ni de los periféricos u otros circuitos conectados a él. En cambio, ISIS puede simular el circuito completo, incluyendo todos los periféricos o circuitos que interactúan con el microcontrolador. Es por eso que Proteus es uno de los simuladores preferidos por quienes diseñan circuitos con microcontroladores. En este programa, los microcontroladores son considerados como un componente más en los circuitos.



Buses

Como adelantamos en el **Capítulo 4**, ahora veremos una función de gran utilidad para dibujar diagramas en el módulo ISIS y, en especial, en circuitos con microcontroladores. Nos referimos a la posibilidad de usar **buses** para realizar la conexión de múltiples líneas, sin tener un exceso de ellas por todo el diagrama.

Etiquetas de líneas de conexión

Es posible asignar un **nombre** o una **etiqueta** a una línea de conexión, para lo cual debemos presionar el botón **Wire Label Mode** de la barra de herramientas de **Modo**. Luego, posicionamos el cursor del mouse sobre una línea de conexión y, cuando este toma la forma de un lápiz de color blanco con una cruz en la punta, hacemos un clic. Aparecerá la ventana llamada **Edit Wire Label**, donde tenemos que escribir un nombre para la línea de conexión y, a continuación, pulsar el botón **OK**. El nombre asignado aparecerá sobre la línea de conexión elegida; esto sirve como una referencia visual o indicaciones especiales. También podemos nombrar una línea si hacemos un clic derecho sobre ella y, en el menú contextual, elegimos la opción **Place Wire Label**.

PARA ASIGNAR UN NOMBRE A UNA LÍNEA DE CONEXIÓN DEBEMOS PRESIONAR WIRE LABEL MODE



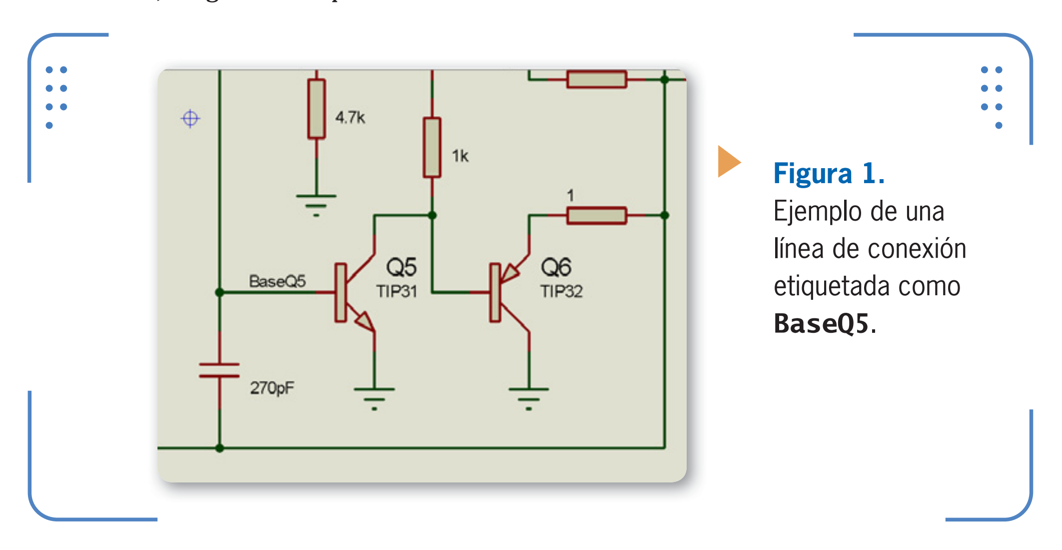


Figura 1. Ejemplo de una línea de conexión etiquetada como **BaseQ5**.

Si queremos retirar el nombre de una línea de conexión, basta con hacer un doble clic sobre la etiqueta y, en la ventana **Edit Wire Label**, borrarlo. Al presionar **OK**, este desaparecerá. Hay que tomar en cuenta que las etiquetas también implican interconexión, si dos o más líneas tienen el mismo nombre. El uso de nombres en las líneas de conexión no solo es útil como referencias visuales, sino que también nos ayudará al momento de usar los buses.

Elementos con pines de bus

Algunos elementos y componentes cuentan con **pines** que son **buses**; se identifican por ser más gruesos que un pin simple y, además, son de color azul. Un ejemplo es el generador de patrones, que tiene un pin de bus llamado **B[0..7]**.

Dibujo de un bus

Un bus es un **grupo de líneas de conexión** representado por una sola línea que es un poco más gruesa que las líneas simples. Para colocar un bus en un circuito, tenemos que presionar el botón **Buses Mode** en la barra de herramientas de **Modo**. Los buses se dibujan de forma similar a las líneas de conexión, con algunas particularidades. Deben conectarse a un pin de bus en algún elemento que cuente con ellos o pueden dibujarse **sin conexión**, es decir, aislados de cualquier elemento o componente.

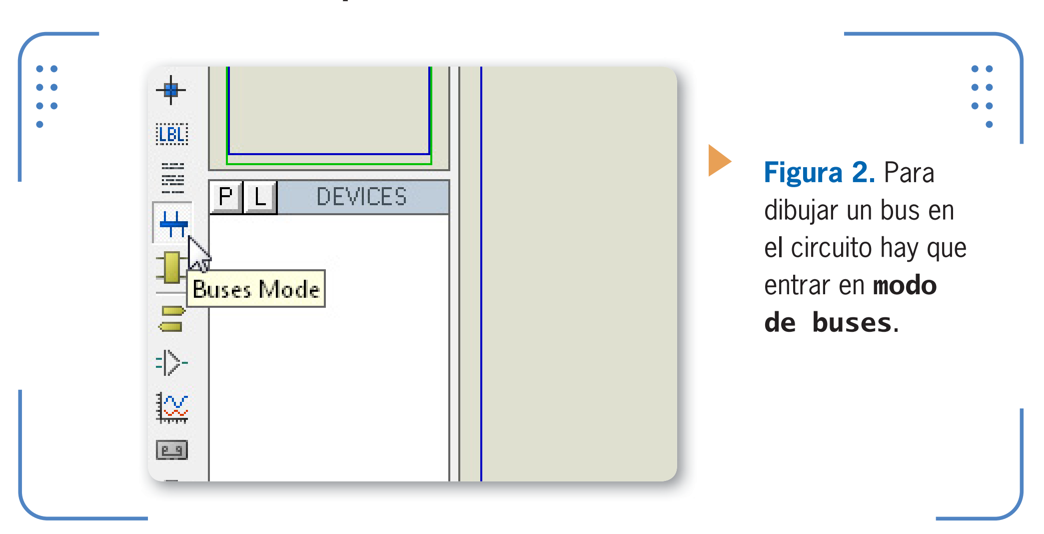


Figura 2. Para dibujar un bus en el circuito hay que entrar en **modo de buses**.

Para agregar un bus a partir de un pin de bus, procedemos de igual forma que con las líneas de conexión simples. Ubicamos el cursor del mouse en el extremo del pin y hacemos un clic para comenzar el dibujo; también podemos iniciarlo en un punto vacío de la ventana de edición, haciendo un clic y comenzando a dibujarlo.

Si necesitamos insertar un cambio de dirección del bus, solo debemos hacer un clic en el lugar donde lo deseamos. Para finalizar,

conectamos el bus a otro pin de bus, a un bus ya existente o, en su defecto, hacemos un doble clic en un lugar vacío. Los buses siguen las mismas reglas de **autoruteo**, y para seleccionar o mover sus segmentos, es tal como en las líneas de conexión simples.

Como podemos apreciar, una vez más, el proceso de dibujo y de conexión de los buses en Proteus es una tarea sumamente sencilla. Solo debemos respetar las especificidades de conexión.

Interconexión de pines usando buses

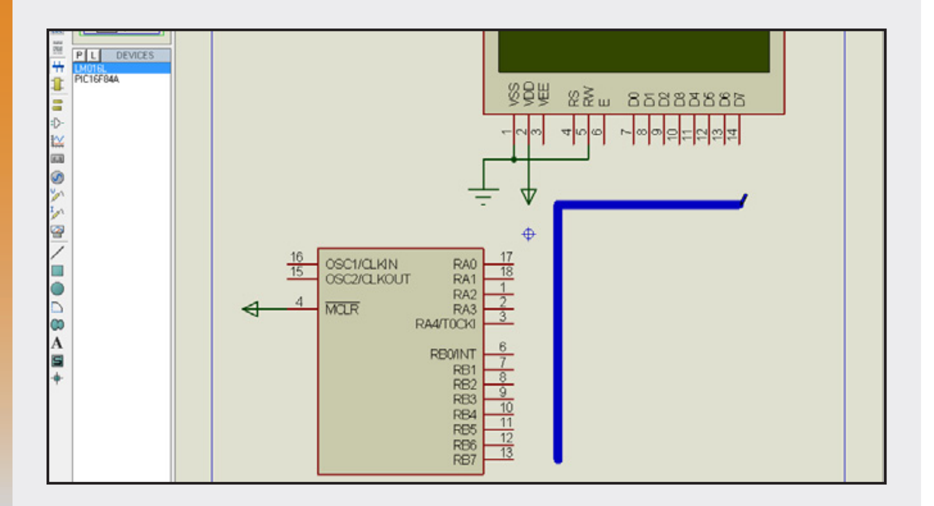
Una vez dibujado el bus, podemos conectarle pines de componentes. Para hacerlo, procedemos de manera usual; al conectar una línea al bus, debemos etiquetarla para que este sepa dónde va esa línea. Para entenderlo mejor, veamos un ejemplo en el siguiente **Paso a paso**.

PASO A PASO: CONECTAR UN LCD A UN PIC

01 Suponga que desea conectar un LCD LM016L a un PIC16F84A mediante un bus. Coloque los componentes en un nuevo diseño y presione el botón Buses Mode para entrar en modo de buses.

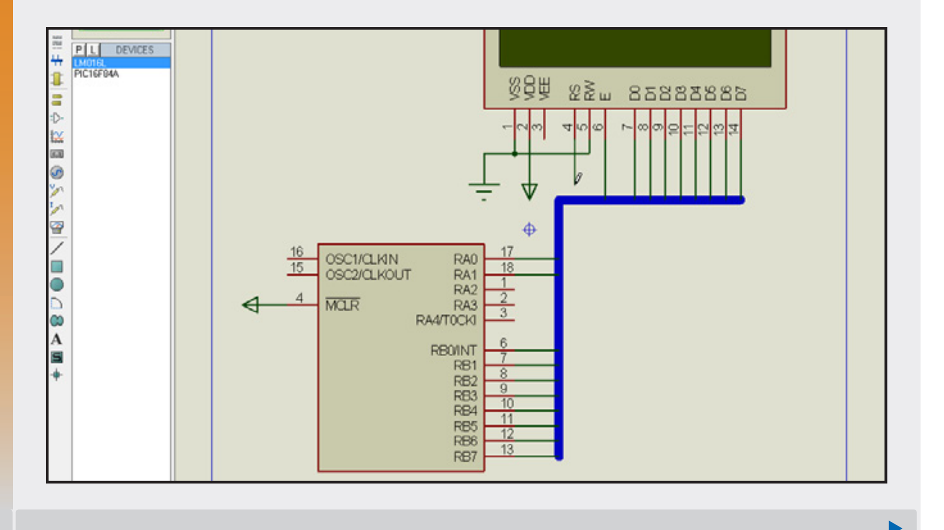
02

Haga un clic para iniciar el dibujo del bus y siga la dirección necesaria hasta terminar. Haga un doble clic en el lugar donde desea finalizar el dibujo.



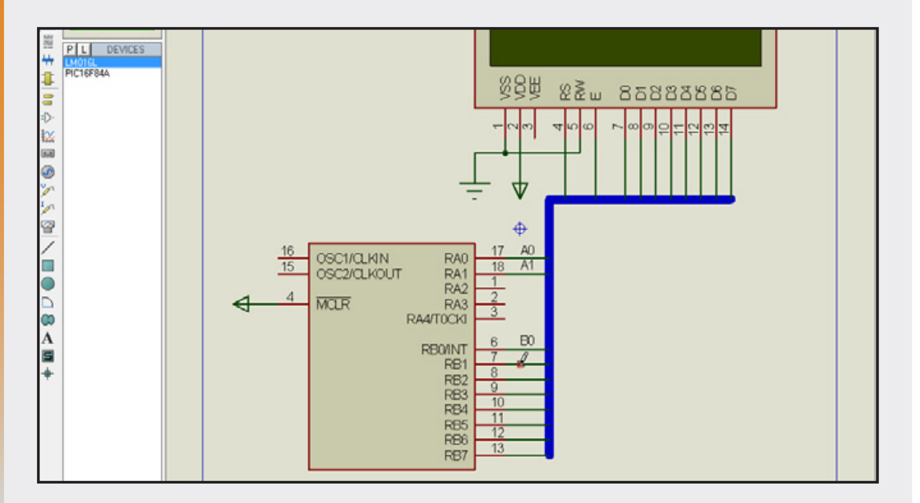
03

Conecte los pines del PIC al bus mediante líneas de conexión simples, y haga lo mismo con los pines del LCD.



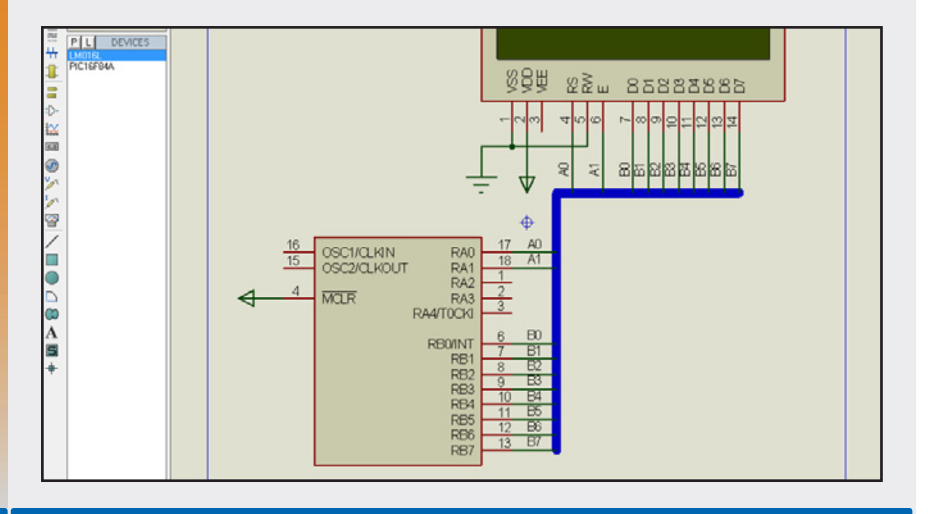


Seleccione `Wire Label Mode` y, luego, la línea que une el pin RA0 del PIC al bus. Se abrirá la ventana `Edit Wire Label`; ingrese un nombre en `String`, por ejemplo A0, y pulse OK. Continúe con los pines restantes, dando nombre a cada línea de conexión; por ejemplo, A0 y A1 para RA0 y RA1, y B0 a B7 para RB0 hasta RB7.





Siguiendo en el modo de etiquetas, seleccione la línea que conecta al pin RS del LCD, en la ventana `Edit Wire Label` haga un clic sobre la f echa de la lista desplegable del campo `String` y elija A0; así, esa línea se conectará a la línea A0 del PIC. Continúe etiquetando las líneas restantes.



Los buses permiten mantener los diagramas ordenados y limpios. Cada línea se identificará en el bus mediante su nombre y se conectará con la línea o las líneas que tengan el mismo nombre o etiqueta.

Etiquetas de bus y terminales de bus

Así como las líneas de conexión pueden ser etiquetadas o nombradas, sucede lo mismo con los buses, aunque en estos se debe especificar un nombre y un rango para las líneas que llevan. Un ejemplo del nombre de un bus es: **D[0..7]**, que representa las líneas llamadas **D0**, **D1**, **D2** hasta **D7**; es un bus de 8 bits. Esto es útil cuando se usan terminales de bus, que, como sabemos, deben tener el mismo nombre para que haya conexión entre ellas.



Propiedades de los microcontroladores PIC

Como todos los componentes de Proteus, los microcontroladores tienen propiedades en las que podemos configurar diversos parámetros; entre ellos, debemos definir el programa que ejecutará. Por lo general, las propiedades de todos los PICs suelen ser similares.

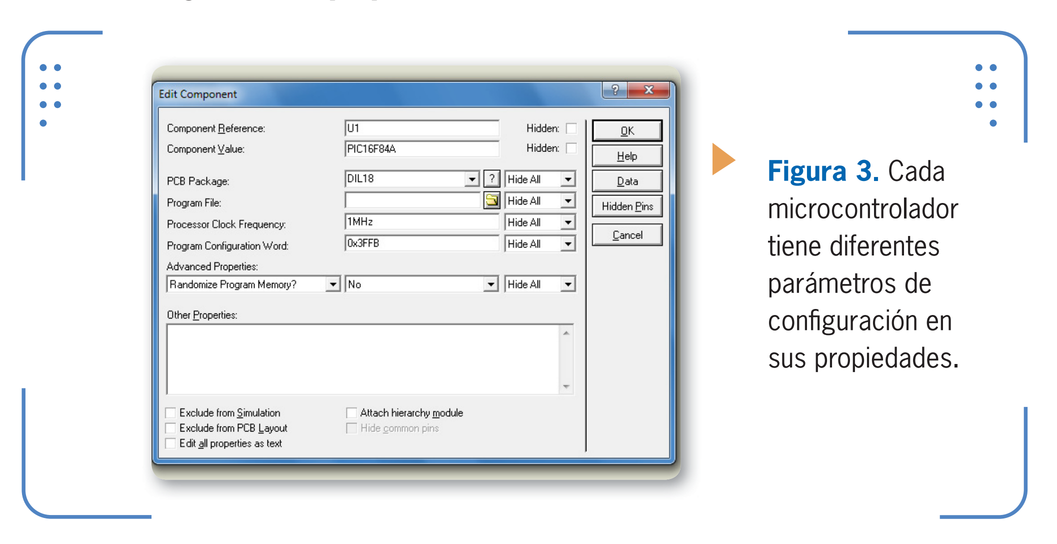


Figura 3. Cada microcontrolador tiene diferentes parámetros de configuración en sus propiedades.

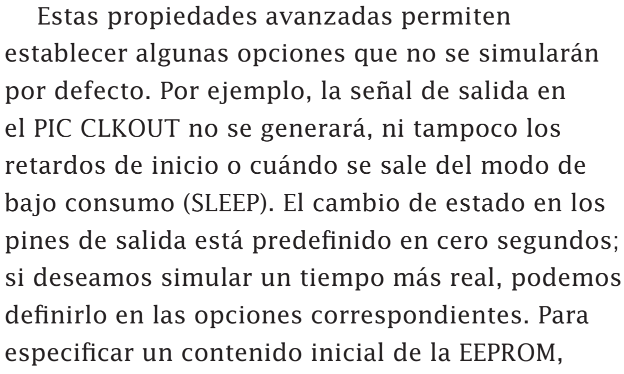
En la **Figura 3** vemos las propiedades de un PIC16F84A, los campos que nos interesan por ahora son:

- **Program Configuration Word**: es la palabra de configuración del PIC. Generalmente no es necesario establecer este valor, ya que lo definirá el propio código máquina que el PIC ejecutará.
- **Processor Clock Frequency**: en este campo debemos determinar la frecuencia de la señal de reloj. Es muy importante definir este valor, porque aunque coloquemos un cristal en el diagrama, la frecuencia de la señal de reloj será la que se ingrese en este campo.
- **Program File**: aquí debemos elegir el archivo que contiene el código o programa que el PIC ejecutará. Podemos trabajar con archivos **.HEX** o **.COF**. La capacidad de hacer depuración depende del archivo usado.
- **Advanced Properties**: para definir algunas propiedades avanzadas, que dependen del PIC utilizado. En algunos tendremos diferentes opciones; las principales se muestran en la **Tabla 1**.

PROPIEDADES DE UN PIC

PROPIEDAD	DESCRIPCIÓN
Model PIC Start-up Delays?	Activa o no la simulación de los retardos de inicio del PIC (POR, PWRT, OST).
Model PIC Wake-up Delays?	Activa o no la simulación de los retardos al salir del modo SLEEP (TOST, TFSCM).
Generate Q Clocks on CLKOUT Pin?	Genera o no la señal de reloj en la salida CLKOUT.
Watchdog Timer Period	Define el período del temporizador Watch Dog Timer.
Data EEPROM Write Delay	Tiempo de retardo de escritura en la EEPROM interna del PIC.
Port Pin Low-High Delay	Retardo en el paso del estado bajo al estado alto en las salidas del PIC.
Port Pin High-Low Delay	Retardo en el paso del estado alto al estado bajo en las salidas del PIC.
Initial contents of EEPROM	Contenido inicial de la EEPROM del PIC.

Tabla 1. Principales propiedades avanzadas de un PIC.



LA PROPIEDADES AVANZADAS ADMITEN OPCIONES QUE NO SE SIMULAN POR DEFECTO

Estas propiedades avanzadas permiten establecer algunas opciones que no se simularán por defecto. Por ejemplo, la señal de salida en el PIC CLKOUT no se generará, ni tampoco los retardos de inicio o cuándo se sale del modo de bajo consumo (SLEEP). El cambio de estado en los pines de salida está predefinido en cero segundos; si deseamos simular un tiempo más real, podemos definirlo en las opciones correspondientes. Para especificar un contenido inicial de la EEPROM, debemos elegir el archivo **.BIN** que contiene los datos, aunque también podemos cargar contenido inicial desde el archivo de programa definido en el campo **Program File**.



Simular con archivos HEX y COF

Si lo que necesitamos es solo simular el funcionamiento del circuito, podemos hacerlo con el archivo **.HEX** generado por el MPLAB o algún otro ensamblador o compilador para PIC. Simplemente, en el campo **Program File** de las propiedades del PIC, elegimos el archivo **.HEX** correspondiente y así simulamos nuestro circuito. Este archivo contiene los datos binarios que el PIC ejecutará, tal como si lo hubiésemos grabado en su memoria de programa.

Muchos ensambladores y compiladores generan también un archivo **.COF**, que puede usarse para depurar el código fuente. Para trabajar con



CONEXIONES CON DOBLE CLIC

Se pueden hacer conexiones rápidas con un **doble clic**. Al tener un bus frente a un grupo de pines de un circuito integrado, por ejemplo, podemos conectar el primero de ellos al bus. En los siguientes haremos **doble clic**, y la conexión al bus se hará de forma automática. Esto permite realizar conexiones a un bus de manera rápida; funciona en todo tipo de conexiones múltiples contiguas.

este tipo de archivos, debemos elegirlo en el campo **Program File** en las propiedades del PIC.

En principio, la simulación continua, es decir, la que se realiza al presionar el botón **Play** de la barra de simulación, será idéntica sin importar si se usa un archivo **.HEX** o **.COF**.

Si optamos por este método para simular o hacer depuración, debemos trabajar con el ensamblador o compilador junto con Proteus. Cada vez que necesitemos hacer un cambio en el código fuente, tenemos que hacerlo desde el ensamblador o compilador; luego ensamblamos o compilamos el programa para generar otra vez el archivo **.HEX** o **.COF**; vamos a la ventana de Proteus y corremos la simulación para ver los resultados de los cambios.



Ensamblar desde ISIS

Junto con Proteus se instalan herramientas para ensamblar y depurar programas desde ISIS, sin tener que instalar ningún ensamblador extra en la computadora. Desde el menú **Source**, podemos configurar las herramientas de ensamblado mediante la opción **Define Code Generation Tools...**. Al seleccionarla, se abrirá la ventana **Add/Remove Code Generation Tools**, donde configuraremos los diferentes ensambladores. En nuestro caso, nos interesa el **MPASMWIN**, que es el ensamblador para PICs.

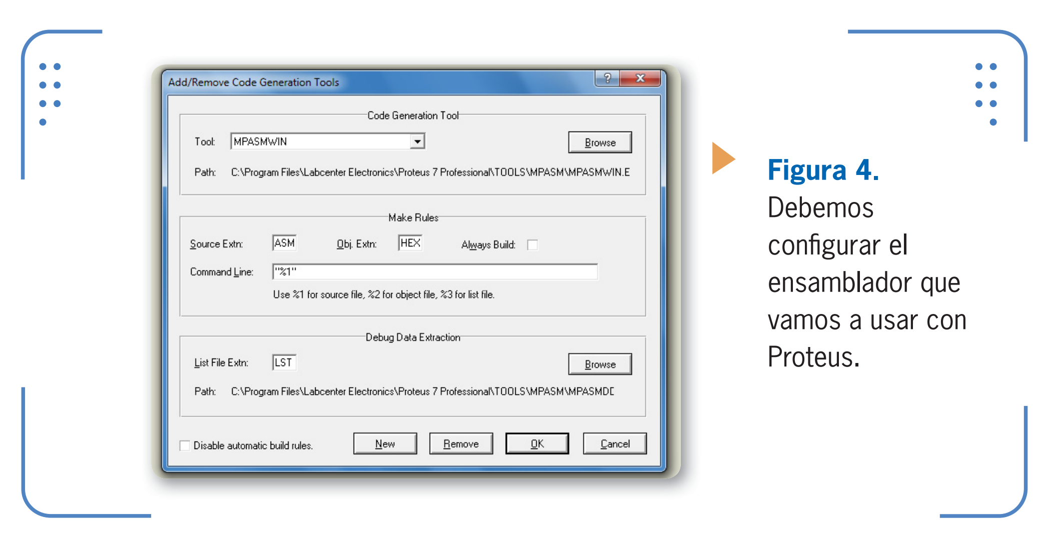


Figura 4. Debemos configurar el ensamblador que vamos a usar con Proteus.

En el campo **Tool**, elegimos la opción **MPASMWIN**. Si no está en la lista, presionamos el botón **New** y buscamos el archivo **MPASMWIN.EXE**, que se encuentra en la carpeta MPASM, la cual, a su vez, está en la carpeta Tools, dentro de la carpeta de instalación de Proteus. En la sección **Debug Data Extraction**, presionando el botón **Browse** y, luego, elegimos el archivo **mpasmddx.exe**, que es el que generará el archivo **.SDI** para la depuración de los programas. Por último, verificamos que esté presente la misma configuración que se muestra en la **Figura 4** y presionamos el botón **OK** para finalizar.

Asignar el código fuente a un PIC

Para ensamblar desde ISIS, debemos asignar el **código fuente** (el archivo **.ASM**) al microcontrolador. Para comprenderlo de forma más clara, veamos un ejemplo real. En el archivo **ContadorPIC.dsn** (y en **ContadorPIC.asm**) tenemos un PIC16F84A que muestra una cuenta en binario en 8 LEDs conectados a su puerto B.

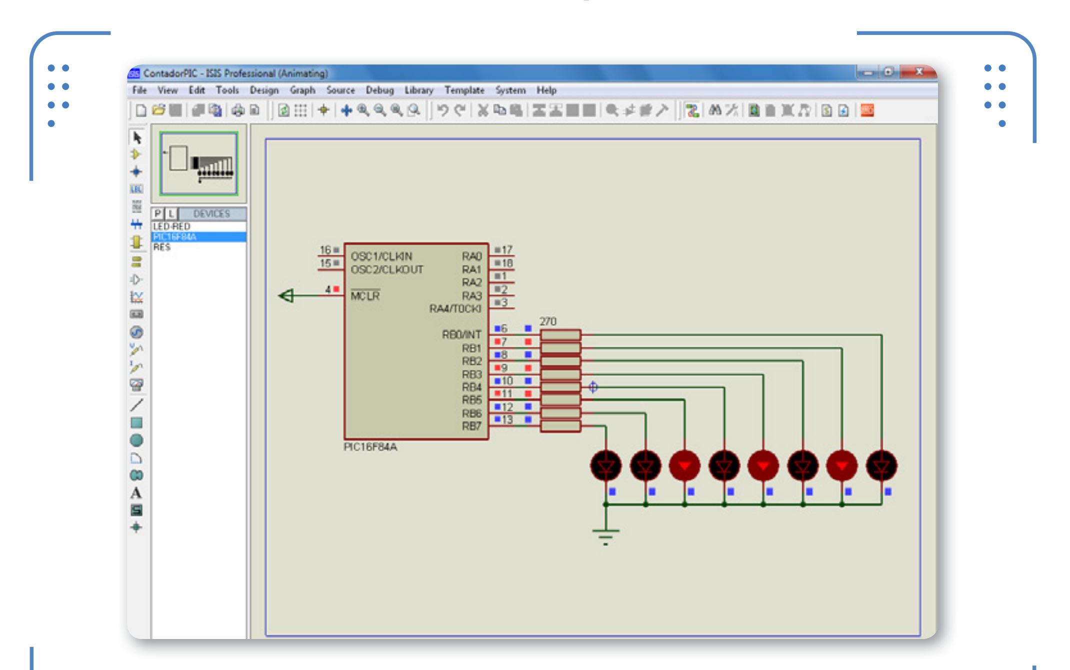


Figura 5. Presentamos un ejemplo de un sencillo contador para aprender a ensamblar desde Proteus.

Para asignar un archivo de código fuente debemos ir al menú **Source/ Add/Remove Source Files...**. En la siguiente **Guía visual**, veremos cómo configurar cada uno de los elementos de esta ventana.

GUÍA VISUAL ■ ASIGNAR CÓDIGO FUENTE A UN PIC

01 **TARGET PROCESSOR:** este campo determina el PIC al cual asignaremos el código fuente; en este caso, el **U1**. Como es posible tener más de un PIC en el circuito, debemos seleccionar el adecuado de esta lista.

02 **CODE GENERATION TOOL:** aquí elegimos el ensamblador que vamos a usar, en este caso, **MPASMWIN,** que ya configuramos antes.

03 **FLAGS:** permite seleccionar algunas banderas para el funcionamiento del ensamblador, las dos principales son:

- **/p[PIC]:** determina el procesador que vamos a usar. Debemos definir el PIC que utilizaremos u obtendremos un error, ya que el ensamblador no sabe a qué PIC se refiere el código fuente. En este caso colocamos: **/pPIC16F84A**.
- **/q[+]o[-]:** define el modo silencioso. Si colocamos **/q+**, no se mostrará la ventana del progreso del ensamblado. Si elegimos **/q-** o no colocamos la bandera, sí se mostrará.

04 **SOURCE CODE FILENAME:** aquí establecemos el archivo de código fuente; podemos seleccionar uno de la lista o cambiarlo con el botón **Change**. Es posible tener más de un archivo asignado a un PIC.



REMOVE: permite quitar un archivo. Al pulsar este botón, se nos pedirá la confirmación, y cuando presionemos **OK**, se lo eliminará de la lista.



NEW: para elegir un nuevo archivo **.ASM**, presionamos este botón y luego localizamos el archivo. Si aún no hay ningún archivo asignado, debemos asignarlo con este botón.

Al asignar un código fuente a un PIC, ya podemos ensamblarlo; para esto solo debemos ir al menú **Source/Build All**, luego se mostrará la ventana con el progreso del ensamblado.

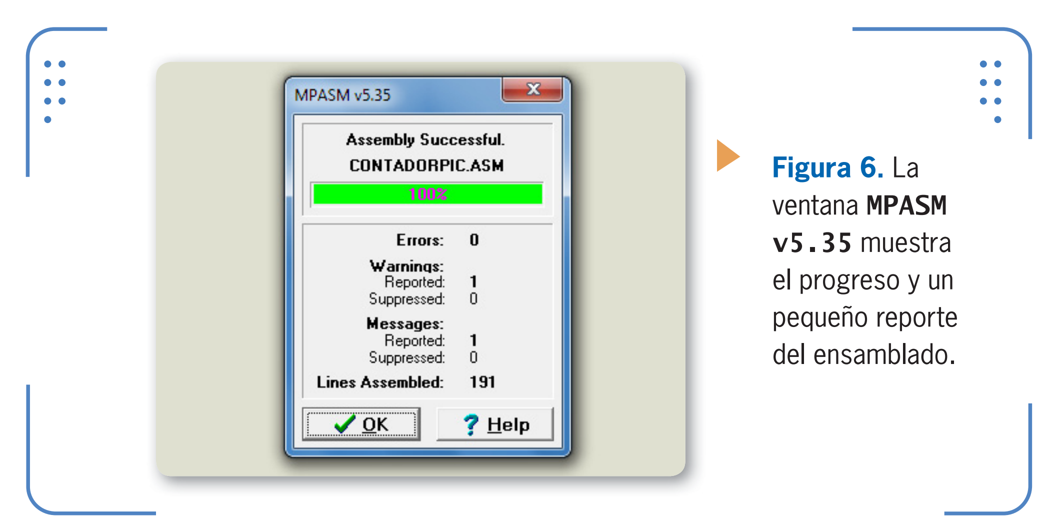


Figura 6. La ventana **MPASM v5.35** muestra el progreso y un pequeño reporte del ensamblado.

Si no se detectan errores, la barra de progreso será de color verde y se mostrará el texto **Assembly Successful**. Luego, al presionar el botón **OK**, se abrirá el informe de simulación con un reporte del



LOS ARCHIVOS DE PROGRAMA

En las propiedades de los microcontroladores PICs es posible especificar los archivos que contienen el programa que ejecutará; los únicos formatos que se pueden colocar en este campo son los archivos **.HEX** y **.COF**. Proteus no es capaz de reconocer directamente un archivo fuente como **.ASM**, aunque puede hacerlo indirectamente a través del **.COF**.

ensamblado. En cambio, si hay algún error durante el ensamblado, la barra de progreso será de color rojo y aparecerá el texto **Errors Found**. Al presionar el botón **OK**, se abrirá el informe de simulación con el detalle completo de los errores.

Al ensamblar sin errores por primera vez, el archivo **.HEX** generado en el proceso se asignará automáticamente al PIC, si es que no existía o no lo habíamos hecho con anterioridad. Si por alguna razón esto no sucede, bastará con hacerlo manualmente. Ya podemos correr la simulación para ver el funcionamiento.

El editor de código fuente SRCEDIT

Proteus cuenta además con un sencillo editor de código fuente en caso de que necesitemos hacer cambios en un código. En el menú **Source** encontramos la lista numerada de los códigos fuente asignados a uno o más PICs; si hacemos clic en alguno de ellos, se abrirá el editor SRCEDIT con el código fuente.

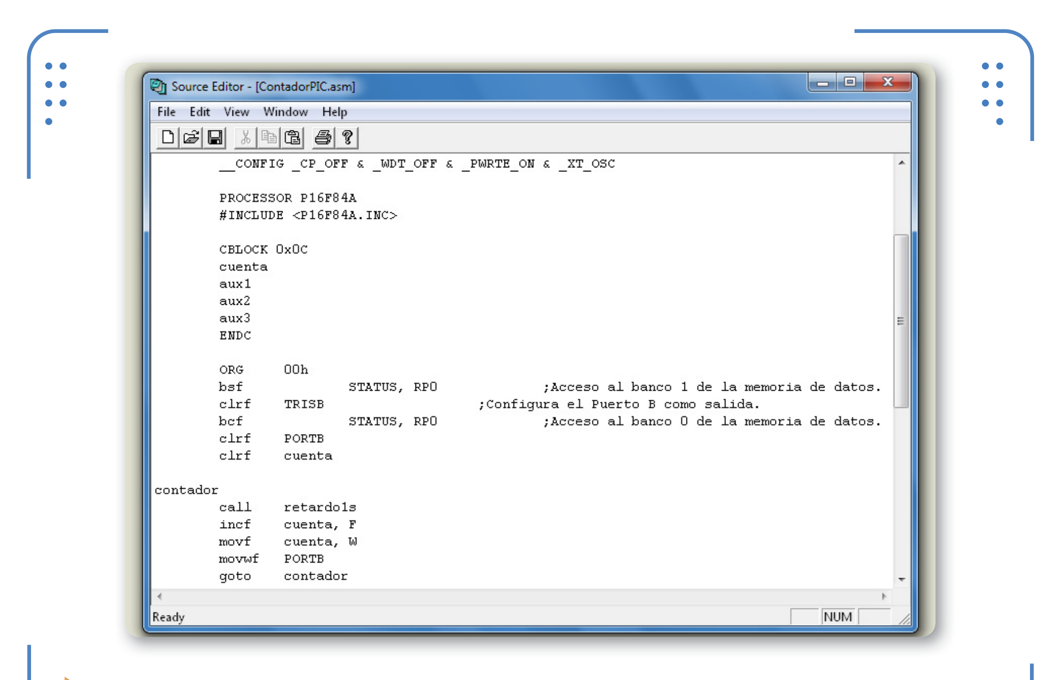


Figura 7. Proteus cuenta con su propio editor de código llamado **SRCEDIT**, que permite realizar modificaciones o correcciones.

El editor no es más que una versión modificada y adaptada del Bloc de notas de Windows. Al abrir un código fuente en él, podemos hacer modificaciones o corregir errores. ISIS detecta automáticamente que se han efectuado variaciones en el editor y, al correr la simulación otra vez, vuelve a ensamblar el código para registrar los cambios. De esta manera, podemos ensamblar y editar nuestros códigos fácilmente desde ISIS y sin necesidad de usar un programa externo.



Depuración en ISIS

Si deseamos **analizar** un programa para realizar cambios o localizar errores, podemos simular su funcionamiento paso a paso, incluyendo una ventana donde se muestra el código fuente. Para llevar a cabo una depuración, debemos estar trabajando con un archivo **.COF**, generado por un ensamblador o compilador externo y agregado al PIC en sus propiedades, o haber ensamblado un código fuente desde ISIS para que **mpasmddx.exe** genere un archivo **.SDI**, que permitirá la depuración.

Una vez que cumplamos con los requisitos anteriores, simplemente presionamos el botón **Pause** o el botón **Step** en la barra de simulación o vamos al menú **Debug/ Start/Restart Debugging**. Con esto se iniciará la simulación paso a paso y automáticamente aparecerá una ventana llamada **PIC CPU Source Code - U1**, que indica que el código pertenece al IC U1. Si estamos trabajando con un archivo **.COF**, se abrirá una ventana llamada **PIC CPU Variables**, que muestra las variables usadas en el programa (solo para lenguajes de alto nivel, como **C**). Esta ventana se llenará de forma automática, si está vacía. Es probable que el archivo **.COF** no soporte la visualización de variables, y esto



¿Y EL CRISTAL?

En la simulación del contador binario con el PIC16F84A, podemos notar que **no hay ningún cristal** conectado al PIC en el diagrama y, sin embargo, la simulación funciona perfectamente. Esto es porque Proteus genera de manera automática la señal de reloj; entonces solo es necesario configurar su frecuencia accediendo a las propiedades del PIC.

depende del compilador que lo generó; si esta ventana está vacía, simplemente podemos cerrarla.

La ventana **PIC CPU Source Code** es donde podemos seguir el funcionamiento del programa, instrucción por instrucción, para localizar errores. Al iniciar la depuración, la primera instrucción del código fuente estará seleccionada, y un pequeño **triángulo de color rojo** indicará que el **contador de programa** se encuentra ahí. Los botones ubicados en la parte superior derecha se usarán para analizar el código de diferentes formas. Veamos más detalles en la siguiente **Guía visual**.

▼ GUÍA VISUAL ■ VENTANA PIC CPU SOURCE CODE


```
2007          __CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC
----
----          PROCESSOR P16F84A
----          #INCLUDE <P16F84A.INC>
----
----          CBLOCK 0x0C
----          cuenta
----          aux1
----          aux2
----          aux3
----          ENDC
----
0000          ORG     00h
0000          bsf             STATUS, RP0          ;Acceso al banco 1 de la
0001          clrf    TRISB                     ;Configura el Puerto B como sali
0002          bcf             STATUS, RP0          ;Acceso al banco 0 de la
0003          clrf    PORTB
0004          clrf    cuenta
----
0005 contador
0005          call    retardo1s
0006          incf    cuenta, F
0007          movf    cuenta, W
0008          movwf   PORTB
0009          goto    contador
----
000A retardo1s
000A          movlw   d'14'
```

01 **CÓDIGO FUENTE:** aquí se muestra el código fuente que ejecutará el PIC; no se puede modificar, es solo para realizar análisis.

02 **ARCHIVOS DE DEPURACIÓN:** aquí se muestra el archivo de depuración utilizado para generar esta ventana; si tenemos más de uno, podemos elegirlo de la lista. ▶

03 **RUN SIMULATION:** este botón corre la simulación de forma continua, es equivalente a presionar el botón **Play**.

04 **STEP OVER SOURCE LINE:** corre la simulación línea por línea del código fuente, sin entrar en subrutinas. Al encontrar una instrucción **call**, se ejecuta toda la subrutina como si fuera una sola instrucción.

05 **STEP INTO SOURCE LINE:** corre la simulación línea por línea entrando en las subrutinas.

06 **STEP OUT FROM SOURCE LINE:** si al estar depurando, hemos entrado en una subrutina, este botón permite salir de ella rápidamente ejecutando el resto en un solo paso. Si presionamos este botón sin estar dentro de una subrutina, el programa se ejecutará de forma continua.

07 **RUN TO SOURCE LINE:** si seleccionamos una línea del código diferente de donde se encuentra el contador de programa y presionamos este botón, el programa se ejecutará de forma continua hasta la línea seleccionada.

08 **TOGGLE BREAKPOINT:** para colocar, desactivar o borrar un punto de ruptura en la línea de código seleccionada.

Podemos usar los botones de la ventana **PIC CPU Source Code** para ejecutar el programa línea por línea y analizar su funcionamiento, o solo una parte de él. En esta ventana no podemos modificar el código fuente; si necesitamos hacerlo, debemos usar el editor SRCEDIT o un editor externo. La ventana **PIC CPU Source Code** solo estará visible al poner pausa; al correr la simulación de forma continua, desaparecerá para poder ver con claridad el circuito y su funcionamiento.



PROTEUS EN MPLAB

Para quienes usan la aplicación MPLAB, tal vez sea útil saber que se puede integrar a Proteus en su interfaz. Simplemente, hay que ingresar al menú **Debugger** (dentro del MPLAB) y, en la opción **Select Tool**, elegir **Proteus VSM**. A continuación aparecerá una ventana con la interfaz de ISIS, donde es posible abrir un circuito y simularlo desde MPLAB.

Puntos de ruptura (breakpoints)

Los puntos de ruptura sirven para detener el flujo del programa en un lugar determinado. Para agregar un punto de ruptura, seleccionamos la línea de código deseada en la ventana **PIC CPU Source Code** y presionamos el botón **Toggle Breakpoint**; al hacerlo, aparecerá un **punto rojo** del lado izquierdo de la línea seleccionada.

Si presionamos otra vez, el punto se convertirá en un **círculo vacío**, lo cual indica que el punto de ruptura se encuentra inactivo. Si volvemos a pulsar el botón, el punto de ruptura se eliminará. También podemos colocar, desactivar o eliminar puntos de ruptura haciendo un **doble clic** sobre cualquier línea del código.

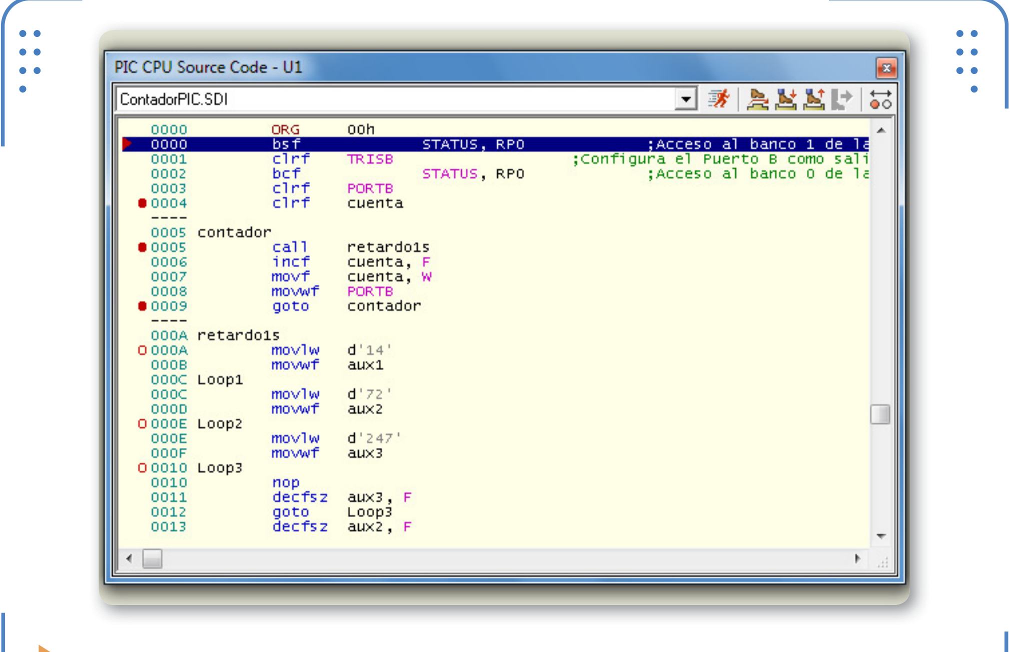


Figura 8. Podemos tener múltiples puntos de ruptura en el código fuente, algunos activos, y otros, inactivos.

Al correr la simulación de forma continua con el botón **Run Simulation** o el botón **Play**, esta se detendrá automáticamente cuando encuentre un punto de ruptura. Esto sirve para observar el funcionamiento de una parte del programa o los valores que toman algunos registros al llegar a ese punto. Es posible colocar todos los puntos de ruptura necesarios.

Si no deseamos que un punto de ruptura actúe, podemos desactivarlo o borrarlo en cualquier momento.

Si queremos correr la simulación sin que los puntos de ruptura actúen y sin tener que borrarlos o desactivarlos, podemos ir al menú **Debug/Execute Without Breakpoints** o presionar **ALT + F12**. También hay una opción llamada **Execute for Specified Time**, que sirve para ejecutar el código durante un tiempo específico. Al seleccionarla, se nos preguntará cuánto tiempo deseamos ejecutar (en segundos) y, cuando pase ese tiempo, la simulación entrará en pausa.

Menú contextual de la ventana PIC CPU Source Code

Si hacemos un clic derecho con el mouse sobre la ventana **PIC CPU Source Code**, veremos un menú contextual con algunas opciones de configuración. En la siguiente **Guía visual** explicaremos cada una de estas opciones en detalle.

GUÍA VISUAL ■ MENÚ CONTEXTUAL

Dissassembly Ctrl+D
Goto Line... Ctrl+L (1)
Goto Address... (2)
Find... Ctrl+F (3)
Find Again Ctrl+G (4)
Toggle (Set/Clear) Breakpoint F9 (5)
Enable All Breakpoints (6)
Disable All Breakpoints (7)
Clear All Breakpoints Ctrl+F9 (8)
Fix-up Breakpoints On Load
Display Line Numbers (9)
Display Addresses (10)
Display Opcodes (11)
Set Font... (12)
Set Colours... (13)

01	**GOTO LINE...:** permite ir a una línea del código. Al pulsar esta opción, se abre una ventana para ingresar el número de la línea correspondiente.
02	**GOTO ADDRESS...:** nos lleva a una dirección del código. Cuando seleccionamos esta opción, se abre una ventana para escribir la dirección deseada.
03	**FIND...:** permite buscar palabras, expresiones o instrucciones dentro del código fuente. Al elegir esta opción, se abre una pequeña ventana para realizar la búsqueda.
04	**FIND AGAIN:** realiza una nueva búsqueda con la última palabra o expresión usada en la opción **Find**.
05	**TOGGLE (SET/CLEAR) BREAKPOINT:** coloca, desactiva o borra puntos de ruptura.
06	**ENABLE ALL BREAKPOINTS:** activa todos los puntos de ruptura existentes.
07	**DISABLE ALL BREAKPOINTS:** desactiva todos los puntos de ruptura existentes.
08	**CLEAR ALL BREAKPOINTS:** borra todos los puntos de ruptura existentes.
09	**DISPLAY LINE NUMBERS:** muestra las líneas numeradas en el código fuente.
10	**DISPLAY ADDRESSES:** muestra las direcciones de las líneas en el código fuente.
11	**DISPLAY OPCODES:** muestra los códigos de operación de cada instrucción del código fuente.
12	**SET FONT:** permite cambiar la fuente que se usa en la ventana.
13	**SET COLOURS:** para cambiar los colores de la ventana.



EL CRISTAL EN EL DIAGRAMA

Como ya sabemos, la señal de reloj del PIC es generada por Proteus de manera automática durante la simulación. Sin embargo, si necesitamos colocar el cristal en el diagrama, podemos hacerlo, y una buena opción podría ser excluir al cristal (y los capacitores) de la simulación en sus respectivas propiedades.

Otras ventanas de depuración

Al comenzar una depuración, tal vez solo aparezca la ventana **PIC CPU Source Code**, pero si vamos al menú **Debug**, encontraremos una lista de otras ventanas disponibles para monitorear la depuración.

Figura 9. El menú **Debug** ofrece otras opciones para la depuración de los programas.

Esta lista numerada de ventanas, ubicada en la parte inferior del menú **Debug**, solo aparecerá si estamos con la simulación en pausa, es decir, en modo de depuración, y puede variar según el PIC usado



MICROPROCESADORES

Además de los microcontroladores, Proteus cuenta con algunos microprocesadores, como, por ejemplo, los siguientes: la familia **68000** de **Freescale**, el **8086** de la familia **i86** de **Intel** y algunos de la familia **Z80**. Aunque, desafortunadamente, el único de estos que tiene modelo, es decir, que puede ser utilizado para simulaciones de circuitos, es el **8086**.

si tenemos más de un microcontrolador o si estamos usando otros periféricos, que también mostrarán algunas ventanas de depuración. En la **Figura 9** vemos la lista del contador que hemos venido estudiando como ejemplo. La opción **1. Simulation Log** aparecerá siempre en cualquier diseño, es la ventana del informe de simulación. Las ventanas visibles tienen una marca (**Figura 9**). Para mostrar una ventana que no tiene marca simplemente hacemos clic en su nombre.

Ventana PIC CPU Registers

La ventana **PIC CPU Registers** muestra los principales registros del PIC, entre los cuales encontramos el **registro de trabajo (W)**, **STATUS**, **PORTA**, **PORTB** y el valor que tienen actualmente, en decimal, hexadecimal y binario. Además, presenta el valor del contador de programa y la instrucción que está a punto de ejecutarse. Es muy útil para monitorear los cambios en esos registros al momento de depurar el programa; los valores se actualizan a medida que ejecutamos el programa línea a línea, los datos que muestra son fijos y dependen del PIC utilizado.

En esta ventana solo tenemos la opción de configurar el tipo de fuente y sus colores a través del menú contextual.

LA VENTANA PIC CPU REGISTERS MUESTRA LOS PRINCIPALES REGISTROS DEL PIC



```
PIC CPU Registers - U1

     PC: $0000                    INSTR.: BSF STATUS,RP0
      W: 198 $C6 %11000110            SP: 0
 STATUS: 28  $1C %00011100           RPX: 0   Z:1 DC:0 C:0
    FSR: 28  $1C %00011100       PCLATH: 0    $00 %00000000
 OPTION: 255 $FF %11111111       INTCON: 0    $00 %00000000
 PORT A: 1   $01 %00000001       TRIS A: 31   $1F %00011111
 PORT B: 135 $87 %10000111       TRIS B: 255  $FF %11111111
```

Figura 10. El contendio de la ventana **PIC CPU Registers** en el ejemplo del contador binario.

Ventana PIC CPU Data Memory

Esta ventana, al activarla, presenta una lista en hexadecimal de todos los registros de datos del PIC, incluyendo los SFR y GPR.

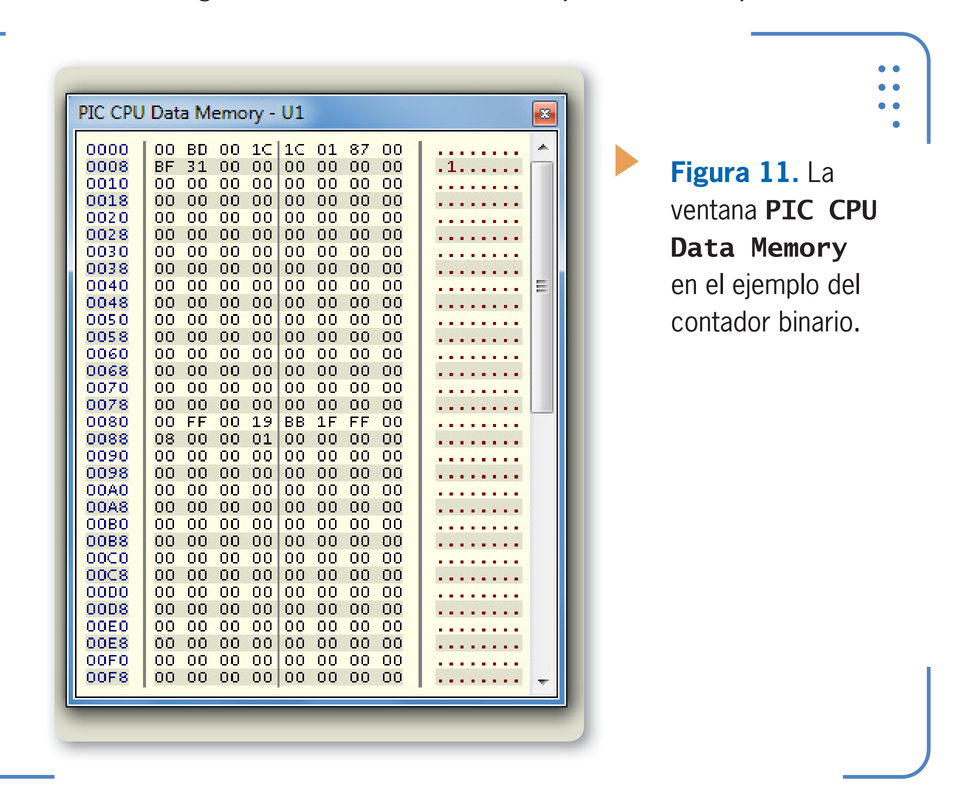


Figura 11. La ventana **PIC CPU Data Memory** en el ejemplo del contador binario.

Al ir recorriendo la ventana con el cursor del mouse, se muestra la dirección, el contenido actual y, entre paréntesis, el contenido previo de cada registro. En la siguiente **Guía visual**, conoceremos en detalle las opciones de su menú contextual.

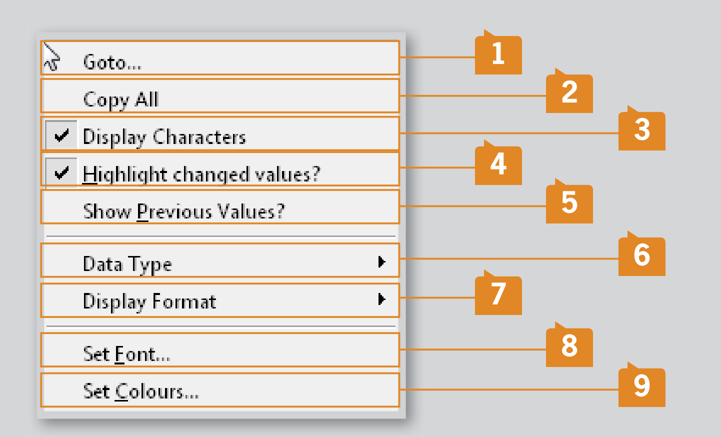


OTROS MICROCONTROLADORES

En el presente capítulo, nos dedicaremos a hablar solo sobre la simulación de microcontroladores PIC, pero es importante saber que la simulación de otros tipos o familias de microcontroladores es muy parecida. Por eso, lo que veremos aquí nos puede servir también como base para la simulación de los otros tipos de microcontroladores que encontramos en Proteus.



GUÍA VISUAL ■ MENÚ CONTEXTUAL



01 **GOTO...:** permite ir a una dirección específica; es particularmente útil en PICs con una gran cantidad de registros de datos.

02 **COPY ALL:** copia todo el contenido de la ventana al Portapapeles.

03 **DISPLAY CHARACTERS:** activa o desactiva la visualización de los datos en ASCII en la parte derecha de la ventana.

04 **HIGHLIGHT CHANGED VALUES?:** permite resaltar con color amarillo oscuro los registros que han cambiado su valor con respecto a la última línea ejecutada; es útil para monitorear fácilmente los cambios.

05 **SHOW PREVIOUS VALUES?:** muestra los valores previos de todos los registros. Al ejecutar una nueva línea de código, esta función se desactiva automáticamente para ver los valores reales en adelante, y no, los previos. Cuando se activa esta función, aparece una ventana que lo advierte.

06 **DATA TYPE:** permite cambiar el tipo de datos en que se muestran los valores de los registros.

07 **DISPLAY FORMAT:** para modificar el formato en que aparecen los valores de los registros.

08 **SET FONT:** para cambiar la fuente que se usa en la ventana.

09 **SET COLOURS:** permite cambiar los colores de la ventana.

Ventana PIC CPU EEPROM Memory

Esta ventana muestra el contenido de la memoria EEPROM interna del PIC. Es útil para monitorear los valores al escribir o leer en ella y comprobar si se hace de forma adecuada.

El menú contextual de esta ventana es idéntico al de la ventana **PIC CPU Data Memory**, que vimos anteriormente.

Ventana PIC CPU Program Memory

La ventana **PIC CPU Program Memory** muestra el contenido de la memoria de programa del PIC.

Al igual que en el caso de la ventana **PIC CPU EEPROM Memory**, el menú contextual de esta ventana también posee las mismas opciones de configuración de la ventana **PIC CPU Data Memory**.

Ventana PIC CPU Stack

Esta ventana muestra el contenido de la pila del PIC y el apuntador. Es útil para comprobar que la pila no se desborde ante algún error o al manejar mal las subrutinas o interrupciones en el programa.

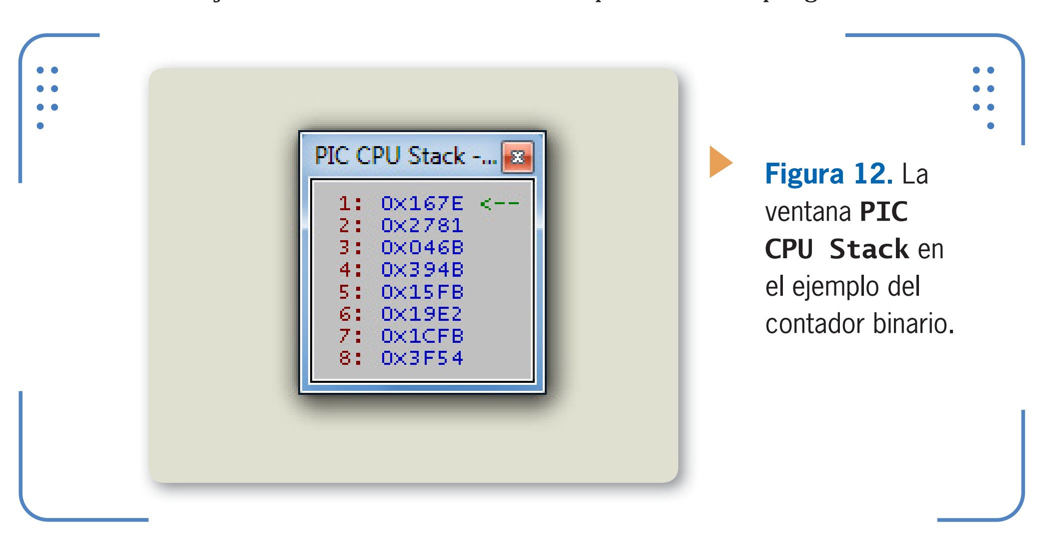


Figura 12. La ventana **PIC CPU Stack** en el ejemplo del contador binario.

Al igual que vimos en la ventana **PIC CPU Registers**, su contenido es fijo, depende del PIC utilizado, y solo es posible configurar el tipo de fuente y los colores que se usarán.



Ventana Watch

Una ventana un tanto especial es la ventana de vigilancia, o **Watch**. Es completamente personalizable y estará visible aun en la simulación corriendo de forma continua. Para abrirla, debemos entrar en modo de depuración y seleccionarla del menú **Debug**. Una vez abierta, estará siempre visible en las simulaciones a menos que la cerremos nosotros.

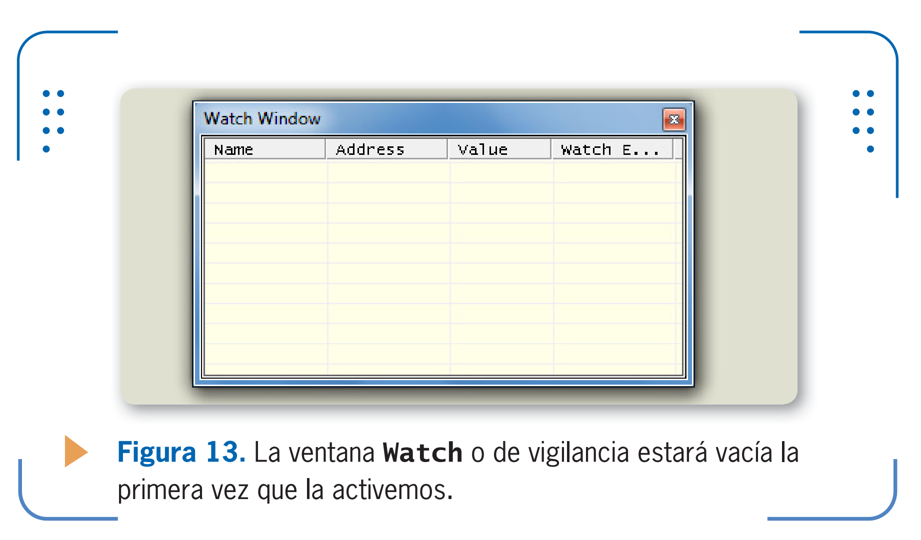


Figura 13. La ventana **Watch** o de vigilancia estará vacía la primera vez que la activemos.

Agregar registros a la ventana Watch

Al abrir esta ventana, estará vacía. Podemos colocar los registros que deseemos para vigilar su comportamiento desde ella y así no tener que controlar todos los de las otras ventanas. Para agregar un registro, simplemente seleccionamos uno o más elementos de las ventanas **PIC**





ARCHIVOS PWI Y VENTANAS DE DEPURACIÓN

Al igual que en los instrumentos virtuales, la posición, tamaño, contenido y visibilidad de las ventanas de depuración en los circuitos con microcontroladores se guardan en el archivo **.PWI** generado al guardar el diseño con el mismo nombre del archivo **.DSN**. Si queremos conservar estas configuraciones, debemos tener siempre el archivo **.PWI** en la misma carpeta donde está el **.DSN**.

CPU Data Memory, **PIC CPU Program Memory** o **PIC CPU EEPROM Memory**, y, a continuación, los arrastramos hacia la ventana **Watch**. Al hacerlo, los registros pasarán a ella, y su nombre será igual a su dirección en la memoria respectiva. En el menú contextual, que se abre cuando hacemos clic con el botón derecho, tenemos diferentes opciones para manejar los registros en la ventana **Watch**, por ejemplo: renombrar un registro (**Rename Item**) o borrarlo (**Delete Item**). También podemos agregar un registro mediante su nombre o dirección; en el menú contextual de la ventana aparecen en la parte superior las opciones: **Add Items (By Name)** y **Add Items (By Address)**.

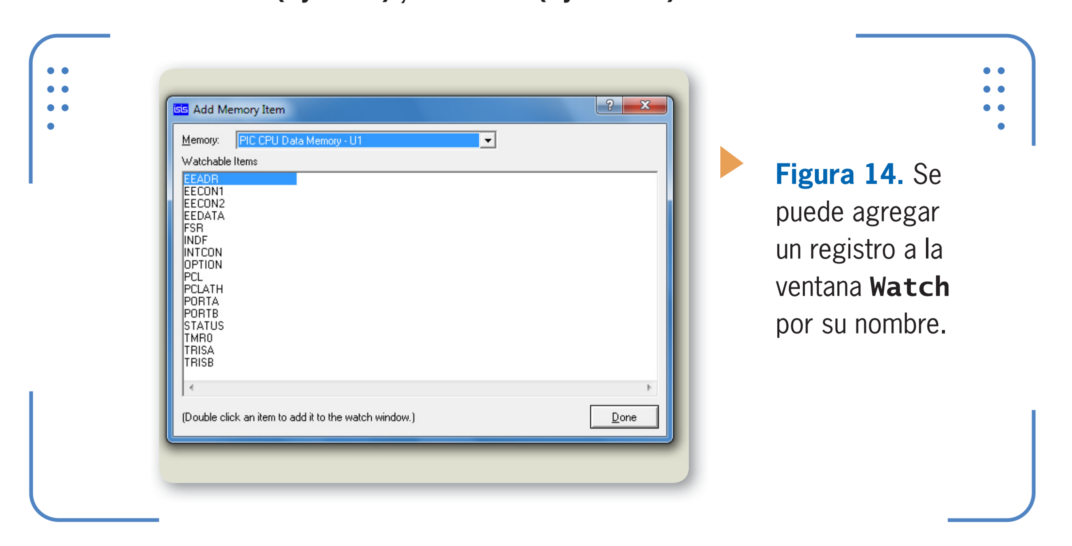


Figura 14. Se puede agregar un registro a la ventana **Watch** por su nombre.

Al elegir la opción **Add Items (By Name)**, aparece la ventana **Add Memory Item**, que se muestra en la **Figura 14**. Allí podemos elegir cualquier registro mostrado en el área **Watchable Items**; con un doble clic en uno de ellos, se agregará a la ventana **Watch**.

AISLAR BITS

La opción **Mask** que encontramos en las condiciones de vigilancia nos permite aislar parte de un registro o incluso un solo bit. Por ejemplo, si deseamos detener la simulación cuando el bit 7 de un registro sea 1, basta con hacer una operación **AND** entre 10000000 (0x80) y el registro deseado, y luego comparar si el valor obtenido es igual (**Equal**) a 10000000.

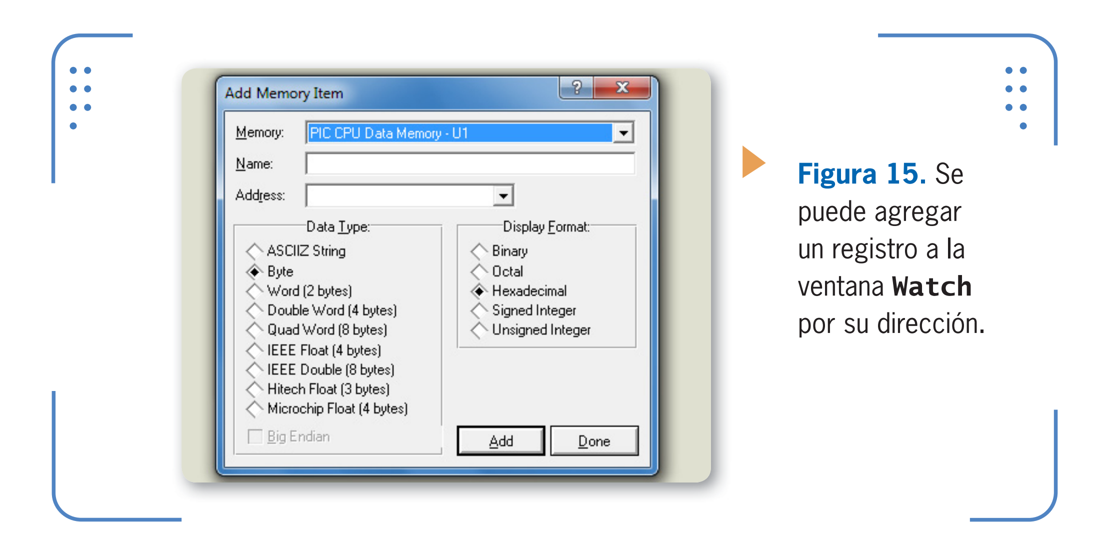


Figura 15. Se puede agregar un registro a la ventana **Watch** por su dirección.

Si seleccionamos la opción **Add Items (By Address)**, se abrirá la ventana **Add Memory Item**, mostrada en la **Figura 15**. En la lista **Memory** elegimos desde qué memoria vamos a agregar los registros. En el campo **Name** definimos un nombre para el registro, así lo tendremos identificado. En la lista **Address** seleccionamos la dirección del registro o la escribimos directamente. Es posible agregar todos los registros necesarios, presionando cada vez el botón **Add** para enviarlos a la ventana **Watch**. Una vez que finalizamos, pulsamos el botón **Done** para cerrar la ventana. Para cada registro podemos elegir el tipo y formato de los datos en los campos **Data Type** y **Display Format**. Así llenaremos la ventana **Watch** solo con los registros que nos interesa vigilar o monitorear durante la simulación o la depuración.

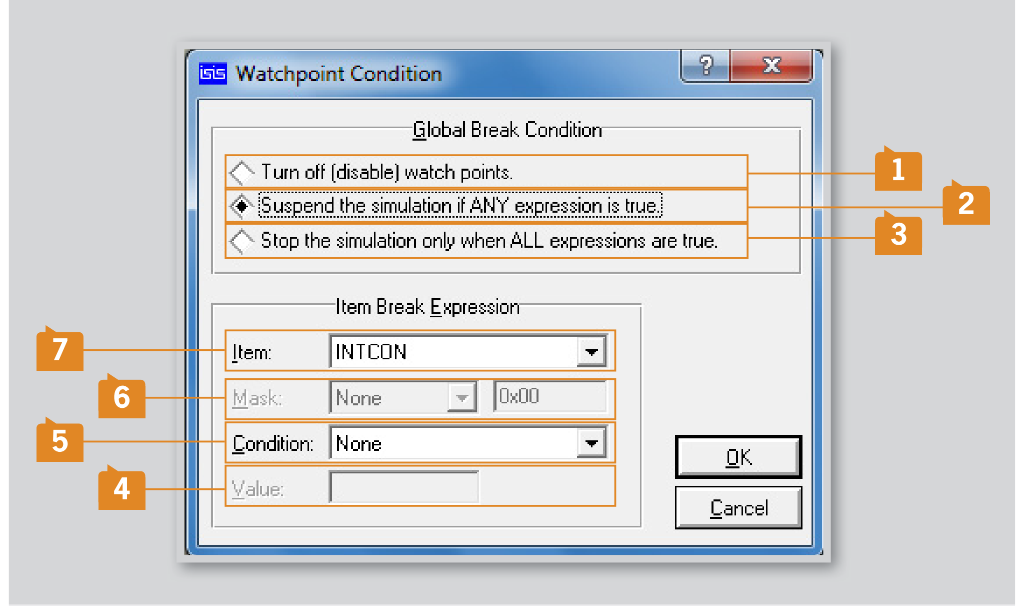
PODEMOS AGREGAR TODOS LOS REGISTROS NECESARIOS PULSANDO ADD

Condiciones de vigilancia

La ventana **Watch** tiene una función de detención al cumplirse una cierta condición en los registros monitoreados. Por ejemplo, podemos hacer que la simulación se detenga cuando uno de los registros tenga un valor concreto. Para agregar una condición, hacemos un clic derecho en la ventana **Watch** y elegimos **Watchpoint Condition** en el menú contextual. En la siguiente **Guía visual** conoceremos estas opciones.

GUÍA VISUAL ■ VENTANA WATCHPOINT CONDITION





01 **TURN OFF [DISABLE] WATCH POINTS:** para desactivar el uso de condiciones de vigilancia.

02 **SUSPEND THE SIMULATION IF ANY EXPRESSION IS TRUE:** permite detener la simulación si cualquiera de las condiciones definidas se cumple.

03 **STOP THE SIMULATION ONLY WHEN ALL EXPRESSIONS ARE TRUE:** para detener la simulación solo si todas las condiciones se cumplen.

04 **VALUE:** establece el valor de la condición; por ejemplo, si en el campo **Condition** elegimos **Equals**, debemos colocar aquí el valor al cual tiene que igualarse el valor del registro.

05 **CONDITION:** establece qué condición debe cumplirse.

- **None:** ninguna.
- **On Change:** cuando haya algún cambio.
- **Equals:** cuando tenga un valor igual a…
- **Not Equals:** cuando el valor sea diferente de…
- **Greater Than:** mayor que…
- **Greater or Equal To:** mayor o igual a…
- **Less Than:** menor que…
- **Less or Equal To:** menor o igual a…



MASK: para hacer una operación lógica AND, OR o XOR entre el registro elegido y el valor que se coloque aquí.



ITEM: para elegir el registro en el cual se creará una condición de vigilancia.

Si establecemos condiciones de vigilancia, al correr la simulación de forma continua (pulsando **Play** o **Run Simulation**), esta se pondrá en pausa automáticamente cuando se cumpla la condición o las condiciones que hemos establecido.

Además, si hacemos un clic derecho sobre la ventana **Watch**, podemos marcar la opción denominada **Show Watch Expressions?** para hacer que dichas condiciones se muestren.

Ejemplo

Veamos un ejemplo del uso de la ventana **Watch**, para lo cual vamos a tomar otra vez el archivo **ContadorPIC.dsn**.

Para abrir la ventana **Watch**, presionamos el botón **Pause**; luego, vamos al menú **Debug** y seleccionamos la opción **2. Watch Window**. Una vez abierta la ventana, hacemos un clic derecho, elegimos la opción **Add Items (By Name)** en el menú contextual que se abre y, luego, hacemos un doble clic en el registro denominado **PORTB**, que es el que nos interesa vigilar.

PODEMOS MARCAR LA OPCIÓN SHOW WATCH EXPRESSIONS? PARA MOSTRAR LAS CONDICIONES







LA VENTANA WATCH Y LOS ARCHIVOS PWI

Los archivos que poseen la extensión **.PWI** tienen una gran importancia para la ventana **Watch**, ya que no solo guardan su posición, tamaño y visibilidad, sino que también almacenan su contenido. Al ser personalizado, siempre debemos tener el archivo **.PWI** junto con el **.DSN** para no perderlo, sobre todo, si hemos colocado muchos registros dentro de ella.

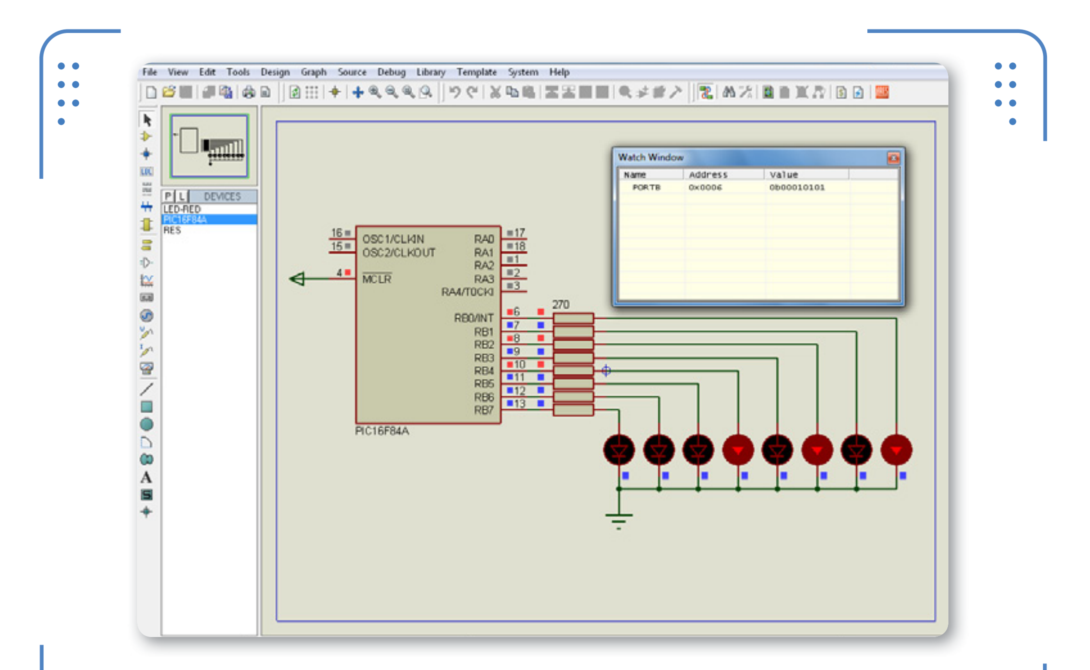


Figura 16. Agregamos el registro del **PORTB** del PIC a la ventana **Watch** para vigilar sus cambios.

A continuación, presionamos **Done**. Al seleccionar **Play** para correr la simulación, observaremos que en la ventana **Watch** se muestran los cambios en el registro del puerto B, con la cuenta en binario.

Ahora supongamos que deseamos detener la simulación cuando el registro **PORTB** tenga un valor de **20** decimal. Para esto, hacemos un clic derecho, en el menú contextual seleccionamos **Watchpoint Condition** y elegimos las siguientes opciones:

- **Suspend the simulation if ANY expression is true**
- **Item**: PORTB (que es el único que tenemos)
- **Mask**: none
- **Condition**: Equals
- **Value**: 20 (en decimal)

Al finalizar, presionamos **OK**. Podemos ver cómo la expresión **= 20** se muestra en la ventana **Watch** en el campo **Watch Expression**.

Ahora, al pulsar **Play**, la simulación correrá, pero cuando el registro **PORTB** alcance el valor **20**, se detendrá automáticamente. De esta

forma, podemos usar la ventana **Watch** y las condiciones de vigilancia para monitorear los registros y detener la simulación ante un evento determinado en los valores de los registros.

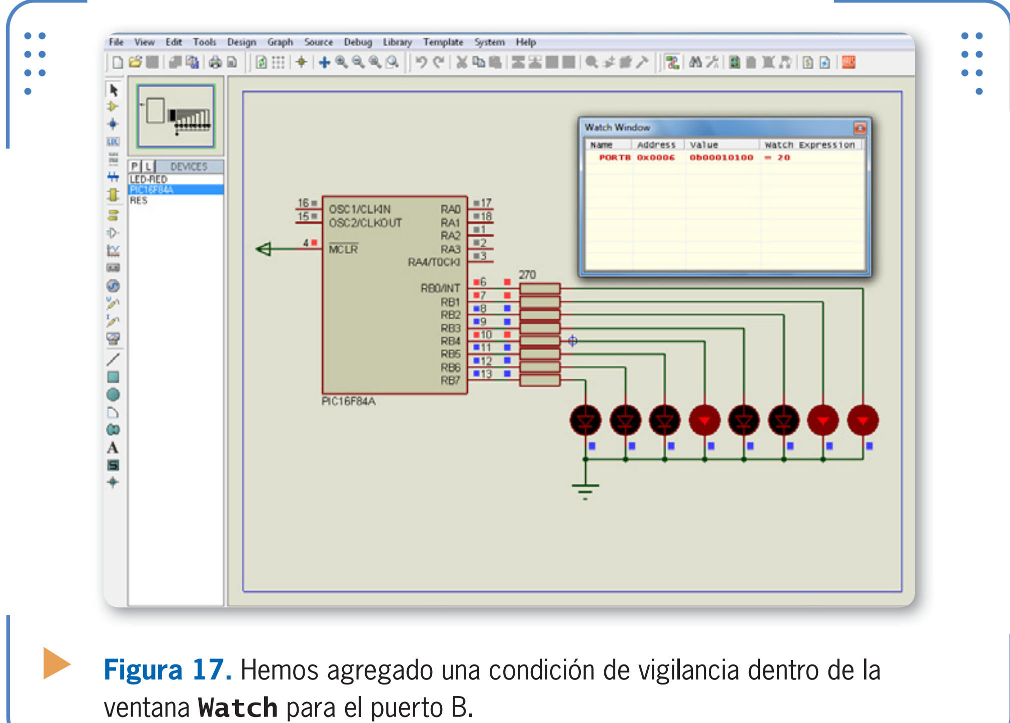


Figura 17. Hemos agregado una condición de vigilancia dentro de la ventana **Watch** para el puerto B.

Animación en el código fuente

Si deseamos hacer una animación automática que corra línea por línea del código fuente, procedemos de la siguiente manera: entramos en modo de depuración presionando **Pause** o **Step**, y en el menú **Debug** elegimos la opción **Animate**. Al hacerlo, la simulación correrá línea por línea del código fuente con animación para seguir el funcionamiento del código de forma automática. La velocidad de la animación estará dada por el parámetro **Step Animation Rate** de las opciones de simulación (**System/Set Animation Options**); aquí elegiremos la velocidad en líneas por segundo; por ejemplo, si colocamos un valor de 5, en cada segundo se ejecutarán 5 líneas. Así podemos establecer una velocidad que nos permita seguir los cambios al ir ejecutándose el programa.

Analizador I2C

En los circuitos con microcontroladores existen herramientas para el análisis de comunicaciones **I2C** y **SPI**. Estos analizadores pueden usarse para monitorear las comunicaciones en tiempo real o para emular un elemento que participa en la transferencia o recepción de datos. Solo estudiaremos el analizador I2C, ya que ambos funcionan de manera muy similar, lo único que cambia es el protocolo.

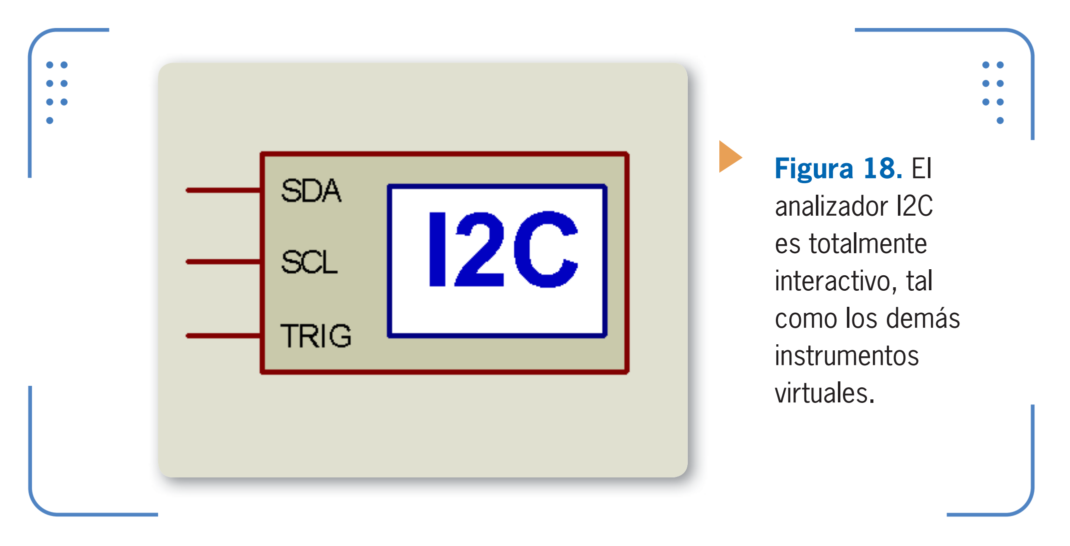


Figura 18. El analizador I2C es totalmente interactivo, tal como los demás instrumentos virtuales.

El analizador I2C tiene tres terminales: **SDA** para la transmisión de datos del BUS I2C, **SCL** para la señal de reloj en el bus I2C, y **TRIG**, que es una terminal de control. Para usar el analizador I2C como monitor, basta con conectar las terminales SDA y SCL a las respectivas terminales en un bus existente, y el analizador mostrará en su pantalla toda la actividad del bus I2C en tiempo real.

Analizador I2C como monitor

Ahora veamos un ejemplo de cómo usar el analizador I2C para monitorear una transmisión en un bus existente.

Debemos descargar y abrir el archivo **PIC-24LC256.dsn**, en el cual podemos observar un circuito con un microcontrolador PIC16F84A y una memoria EEPROM 24LC256.

Figura 19. El analizador I2C como **monitor** en una conexión PIC16F84A con una memoria 24LC256.

Si abrimos el archivo y corremos la simulación, veremos cómo se abre la ventana **I2C Debug** y en ella se muestra en detalle la transmisión. En este ejemplo, se escribe una serie de datos en la memoria 24LC256 a través del bus I2C. Al finalizar la escritura, se leen los datos y se envían al LCD para verificar que se escribieron correctamente. Es posible leer en la primera línea del analizador el tiempo de inicio y fin de la escritura, que es el primer grupo de envío de datos (28.538 ms a 30.141 ms), y después, todo el protocolo de transmisión comenzando por **Start** (S), para finalizar con **Stop** (P).



TRANSMISIÓN Y RECEPCIÓN I2C

Las flechas de color **azul** en el analizador I2C representan recepción, y las de color **rosa**, transmisión. Si se detecta una secuencia no válida, en lugar de una flecha, aparecerá un signo de interrogación. En el informe de simulación habrá más datos en caso de secuencias no válidas o violaciones al protocolo en el bus.

ANALIZADOR I2C

SÍMBOLO	DESCRIPCIÓN
S	Condición Start
A	Bit de reconocimiento (ACK)
Sr	Condición Start repetido
N	Bit de reconocimiento negado (NACK)
P	Condición Stop

Tabla 2. Símbolos en el analizador I2C.

Además, los números representan los datos transmitidos a través del bus en hexadecimal. Si hacemos un clic en los signos más (**+**),veremos con mayor detalle la transmisión de cada condición o dato, incluso, de cada bit de un dato enviado. En la segunda línea que comienza en 35.338 ms tenemos la transmisión al momento de la lectura de la EEPROM. Hemos agregado la ventana **Watch** para verificar los datos de la EEPROM. Si pausamos la simulación, veremos además la ventana **I2C Memory Internal Memory - U2**, que muestra la memoria del 24LC256; también está disponible en el menú **Debug**.

Datos persistentes

En el ejemplo que recién vimos, al correr la simulación una segunda vez (podemos presionar **Pause**), notaremos que los datos en la memoria 24LC256 ya están escritos en ella. Tal como en la realidad, los datos en este tipo de memorias **no volátiles** (la EEPROM interna del PIC es otro ejemplo) quedarán grabados hasta que se cambien. Esto puede ser conveniente en algunos casos, pero no en otros. Si necesitamos borrar los datos en este tipo de dispositivos, podemos ir al menú **Debug/Reset Persistent Model Data** antes de iniciar una simulación, y con esto se borrarán todas las memorias no volátiles que tengamos en el circuito. Podemos hacer esto para comprobar que nuestro ejemplo realmente graba los datos en la EEPROM.

Analizador I2C como maestro

Para usar el analizador I2C como dispositivo maestro, debemos establecer la secuencia de comunicación. Podemos hacerlo desde sus propiedades o de forma interactiva. En la siguiente **Guía visual** conoceremos en detalle algunas de las propiedades.

GUÍA VISUAL ■ PROPIEDADES DEL ANALIZADOR I2C

Edit Component

Component Reference: Hidden:
Component Value: Hidden:
OK
Help
Cancel
1 Clock frequency in Hz: 100000 Hide All
2 Address byte 1 (Default) Hide All
Address byte 2 (Default) Hide All
3 Stop on buffer empty? Hide All
Advanced Properties:
4 Time display precision (Default) Hide All
Other Properties:
5 {SEQUENCE000=S}
{SEQUENCE001=0xA0}
{SEQUENCE002=0x00}
{SEQUENCE003=0x00}
{SEQUENCE004='a'}
{SEQUENCE005='b'}
Exclude from Simulation
Attach hierarchy module
Exclude from PCB Layout
Hide common pins
Edit all properties as text

01 **CLOCK FREQUENCY IN HZ:** define la velocidad del bus I2C; por defecto, es 100 KHz.

02 **ADDRESS BYTE 1 Y ADDRESS BYTE 2:** en caso de usar el analizador como esclavo, aquí se define su dirección en el bus, ya sea de 7 bits con solo la opción 1 o de 10 bits con ambas.

03 **STOP ON BUFFER EMPTY?:** para detener (pausar) la simulación si el buffer de datos ha quedado vacío.

04 **ADVANCED PROPERTIES:** para establecer las propiedades avanzadas:

- **Time display precision:** define el número de decimales en los valores del tiempo al monitorear la conexión; por defecto es 3, y puede ser hasta 10.
- **New line after:** determina el número de datos que se muestran por línea en la ventana de monitoreo; por defecto es 64.
- **Queue stored sequences at startup:** para elegir si las secuencias almacenadas se emiten al iniciar la simulación o no.
- **Sequence file:** establece un archivo de texto que contiene secuencias. ▶



OTHER PROPERTIES: en este campo se define la secuencia deseada que se enviará, además de otras propiedades.

Se puede establecer una secuencia completa de transmisión simplemente escribiéndola, por ejemplo:

```
S 0xA0 0x00 0x00 0x32 P
```

De esta manera, se enviará toda la secuencia en un solo paso. También se puede definir la secuencia en forma de lista, por ejemplo:

```
S
0xA0
0x00
0x00
0x32
P
```

Así se enviará la secuencia línea por línea; además, podemos hacer una lista de secuencias completas. Es posible definir una secuencia de datos desde la ventana interactiva, presionamos **Pause** para acceder a ella y construimos las secuencias. Los datos de las secuencias se pueden escribir en diferentes formatos.



DEPURACIÓN EN ISIS CONTRA MPLAB

Si hemos realizado depuración de programas en MPLAB usando el simulador MPLABSIM, entonces seguramente estaremos familiarizados con los botones de depuración de la barra de herramientas **Debug** (de MPLAB). Los botones de la ventana **PIC CPU Source Code** en ISIS son muy parecidos y entonces veremos que será fácil aprender su funcionamiento.





DATOS DEL ANALIZADOR I2C

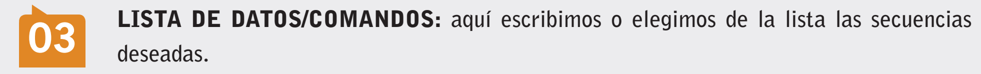
FORMATO	ESCRITURA
Binario	%_ o _b ejemplos: %01001101, 01001101b
Decimal	_d, ejemplo: 99d
Hexadecimal	0x_, $_ o _h, ejemplos: 0x32, $32, 32h
ASCII	'_', ejemplo: 'S'

Tabla 3. Formato para datos en el analizador I2C.

GUÍA VISUAL ■ VENTANA DEL ANALIZADOR I2C

01 **VENTANA DE MONITOREO:** aquí aparece toda la actividad del bus I2C en tiempo real.

02 **PREDEFINED SEQUENCES (SECUENCIAS ALMACENADAS):** aquí se almacenan las secuencias que definiremos.



03 **LISTA DE DATOS/COMANDOS:** aquí escribimos o elegimos de la lista las secuencias deseadas.

04 **DELETE:** para borrar una o más secuencias o datos, seleccionados en la ventana de secuencias almacenadas.

05 **ADD:** para agregar la secuencia, comando o dato escrito en la **Lista de datos/comandos** a la sección de secuencias almacenadas.

06 **QUEUE:** con este botón las secuencias almacenadas pasan al campo **Queued Sequences** y de ahí se envían inmediatamente al bus, si la simulación está corriendo.

07 **QUEUED SEQUENCES:** este es el buffer donde las secuencias se envían al bus en cuanto es posible.

Podemos enviar las secuencias almacenadas de forma manual, mediante alguno de estos procedimientos en la ventana interactiva:

- Seleccionamos alguna y presionamos el botón **Queue**.
- Hacemos un doble clic sobre ella.
- Seleccionamos una secuencia en la **Lista de datos/comandos** y presionamos **Queue**.

Si queremos enviar secuencias automáticamente, podemos usar la terminal **TRIG**; cada vez que se dé un flanco de subida en ella, una secuencia almacenada será enviada por el bus.

En el archivo **I2C-Maestro.dsn** tenemos un ejemplo de la manera en que se usa el analizador I2C como maestro. Hemos reemplazado el PIC con un analizador I2C, y desde él enviamos las secuencias necesarias para escribir las ocho primeras letras del alfabeto en la EEPROM. Se ha definido la secuencia en forma de lista, y el envío se hace automáticamente conectando un generador en la terminal **TRIG**.



CAMBIAR SRCEDIT

Si el editor de Proteus, **SRCEDIT**, no es de nuestro agrado, podemos cambiarlo por otro editor de texto, como **UltraEdit**. El nuevo editor debe soportar comandos DDE. Para cambiarlo vamos al menú **Source/ Setup External Source Editor...**; se abrirá una ventana para localizar el ejecutable del editor deseado.

Figura 20. Podemos utilizar el analizador I2C como dispositivo **maestro** para escribir en la EEPROM.

Podemos ver en este ejemplo cómo efectivamente se escriben los datos en la EEPROM sin necesidad de un microcontrolador. La simulación se pone en pausa de manera automática con una condición de vigilancia (no nos olvidemos de borrar los datos persistentes).

Analizador I2C como esclavo

Para usar el analizador I2C como esclavo solo hace falta definir su dirección como tal en los campos respectivos, así responderá ante las





LA SALIDA DE SECUENCIAS

El buffer de secuencias del analizador I2C es un lugar donde se almacenan temporalmente los datos que están a punto de enviarse por el bus I2C. Esto se hace tan pronto como sea posible; si la velocidad con que las secuencias salen es muy alta, tal vez no observemos nada en esta sección del analizador.

peticiones del maestro en esa dirección. Debemos indicar también una secuencia de respuesta, que dependerá del dispositivo que intentemos emular. En el archivo **I2C_Esclavo.dsn** tenemos un nuevo ejemplo; esta vez, hemos sustituido la memoria 24LC256 por un analizador I2C que la emulará como esclavo.

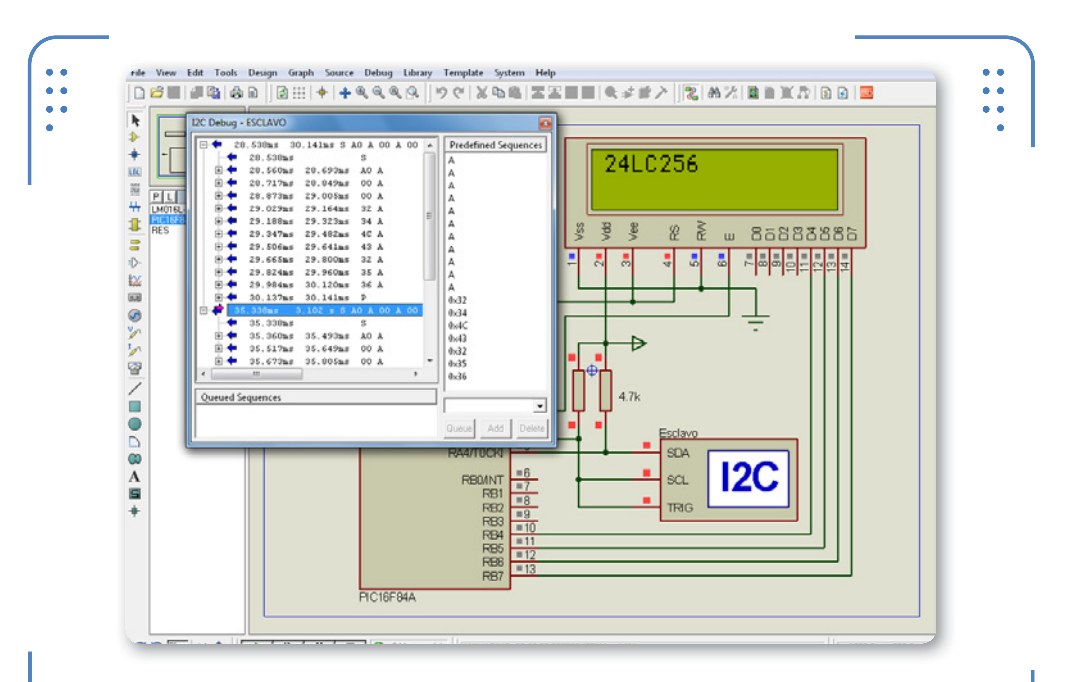


Figura 21. Podemos utilizar el analizador I2C como dispositivo **esclavo** emulando a la memoria 24LC256.

Hemos conectado la terminal **TRIG** a la línea **SCL** para que la señal de reloj pueda disparar las secuencias de respuesta del analizador cuando se requiera. Luego, debemos definir una secuencia de



MÁS SOBRE LOS ARCHIVOS COF

Los archivos **.COF** enlazan el archivo fuente **.ASM** para la depuración, por lo que es necesario tenerlo en la misma carpeta. Si no se encuentra el **.ASM**, la ventana **PIC CPU Source Code** estará vacía con el mensaje: **No source line at PC address (PC=0000)** y no se hará la depuración.

respuestas que emula a la memoria 24LC256, de tal modo que el circuito completo funciona exactamente igual que si la memoria estuviera ahí. Colocamos los datos que se van a leer de forma manual, ya que, obviamente, el analizador recibe los datos del PIC, pero como no es una memoria, no los escribirá en su interior, y no habrá lectura real de datos al intentar leer la memoria emulada.



RESUMEN

Una de las ventajas de Proteus es la capacidad de simular circuitos que contienen diversos tipos de microcontroladores, con la posibilidad de simular perfectamente estos dispositivos con todas sus funciones. Además, es posible efectuar una depuración para localizar errores o realizar cambios en los códigos fuente. En este capítulo, hemos estudiado cómo se simulan los circuitos que contienen microcontroladores PIC, cómo depurar un código fuente y cómo trabajar completamente desde Proteus, sin necesidad de un ensamblador externo.

Actividades

TEST DE AUTOEVALUACIÓN

1 Mencione dos tipos de microcontroladores que puede simular Proteus.

2 ¿Qué es un bus?

3 ¿Para qué se utilizan los buses?

4 ¿Cómo se configura la frecuencia de reloj en un PIC?

5 ¿Cómo se define el archivo que el PIC ejecutará?

6 ¿Qué formatos de archivos se admiten en las propiedades de un PIC?

7 ¿Cómo se asigna un código fuente a un PIC?

8 ¿Cómo se ensambla un código fuente en ISIS?

9 ¿Cómo se inicia una depuración?

10 ¿Cómo se abre la ventana **Watch**?

ACTIVIDADES PRÁCTICAS

1 Busque un circuito que contenga un microcontrolador PIC, dibújelo en ISIS y simúlelo para observar su funcionamiento.

2 Abra el archivo **ContadorPIC.dsn** y modifique el código fuente para que la cuenta se haga en sentido contrario (de 11111111 hasta 00000000). Realice los cambios desde el editor de Proteus, sin usar programas externos.

3 Abra el archivo **I2C-Maestro.dsn** y agregue las secuencias necesarias en el analizador I2C para escribir el alfabeto completo en la memoria 24LC256.

4 Abra el archivo **I2C-Maestro.dsn** y reemplace el generador (disparo) por un LOGICTOGGLE a la entrada TRIG del analizador I2C.

5 Una vez colocado el LOGICTOGGLE simule el circuito, esta vez usted dará los pulsos con él, para disparar las secuencias manualmente en el analizador I2C.



Manejo de diagramas y componentes

Proteus no solo permite simular circuitos electrónicos, también dispone de herramientas para el manejo de los diagramas. Mediante ellas podremos mantener los circuitos ordenados y trabajar en más de una hoja por diseño o crear diseños con subcircuitos. Además, construiremos nuevos símbolos de componentes o editaremos los que ya existen para adaptarlos a nuestras necesidades.



Diagramas en múltiples hojas

Cuando dibujamos un diagrama en una hoja de papel, puede suceder que el espacio no sea suficiente y tengamos que dividir el dibujo en más de una de ellas. En Proteus por supuesto que también existe la posibilidad de distribuir el diagrama en varias hojas de trabajo. De hecho, al iniciar un nuevo diseño, lo que estamos haciendo es, simplemente, utilizar solo una hoja, hasta ahora.

Si vamos al menú **Design** y seleccionamos la opción **New Sheet**, se crea una nueva hoja vacía, donde podemos colocar un circuito o parte del que queremos dividir. Es posible agregar tantas hojas como necesitemos. En la parte inferior del mismo menú, aparece una lista numerada de todas las hojas que contiene el diseño, y se muestra una marca en la que se encuentra activa, es decir, sobre la hoja de trabajo que estamos viendo en ese momento. Basta con hacer un clic sobre cualquier hoja de la lista para ir a ella.

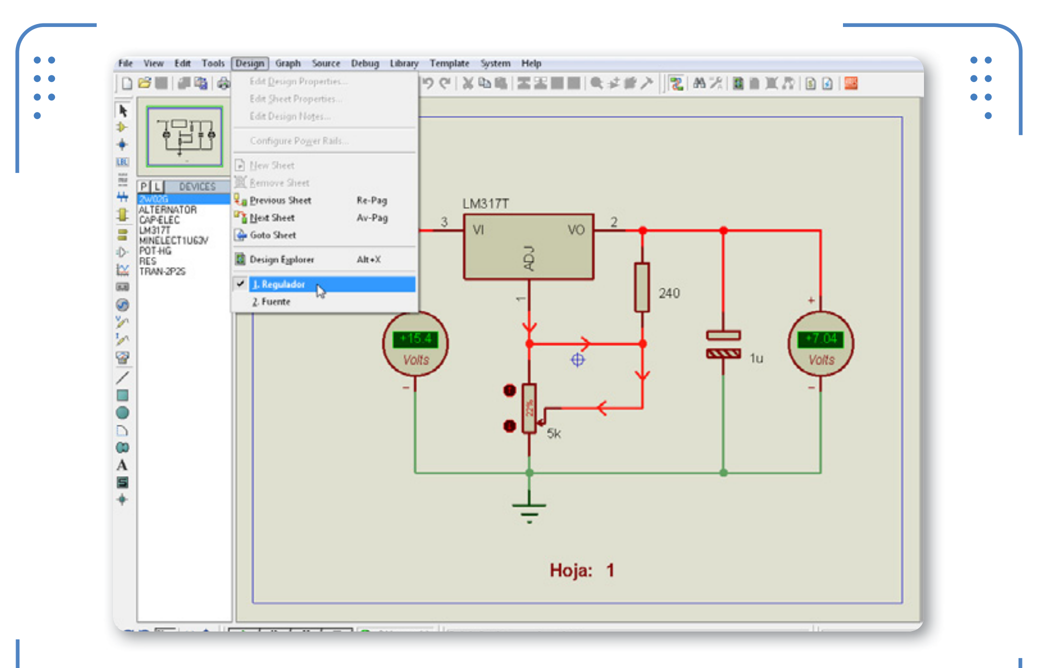


Figura 1. Tenemos la primera hoja del archivo **Fuente.dsn**, donde podemos observar el regulador variable.

Veamos un ejemplo de un diseño con más de una hoja. Utilizaremos el archivo **Fuente.dsn**, en el que tenemos una fuente de voltaje con un regulador LM317 para obtener una fuente regulada variable. El diseño está dividido en dos hojas: en la 1 tenemos la parte del regulador variable, y en la 2, la fuente lineal que alimenta el regulador. Como en este caso solo queremos dar énfasis a la sección del regulador, es por eso que la fuente está en otra hoja. En el menú **Design** aparecen las dos hojas del diseño, llamadas **1. Regulador** y **2. Fuente**.

Para navegar entre las diferentes hojas de un diseño, también podemos hacerlo con las opciones **Previous Sheet** o **Next Sheet** del menú **Design**, o presionando en el teclado **RE PÁG** o **AV PÁG**. La opción **Goto Sheet** del menú **Design** abrirá un cuadro de diálogo con la lista de las hojas del diseño; al elegir una y presionar **OK**, accederemos a ella.

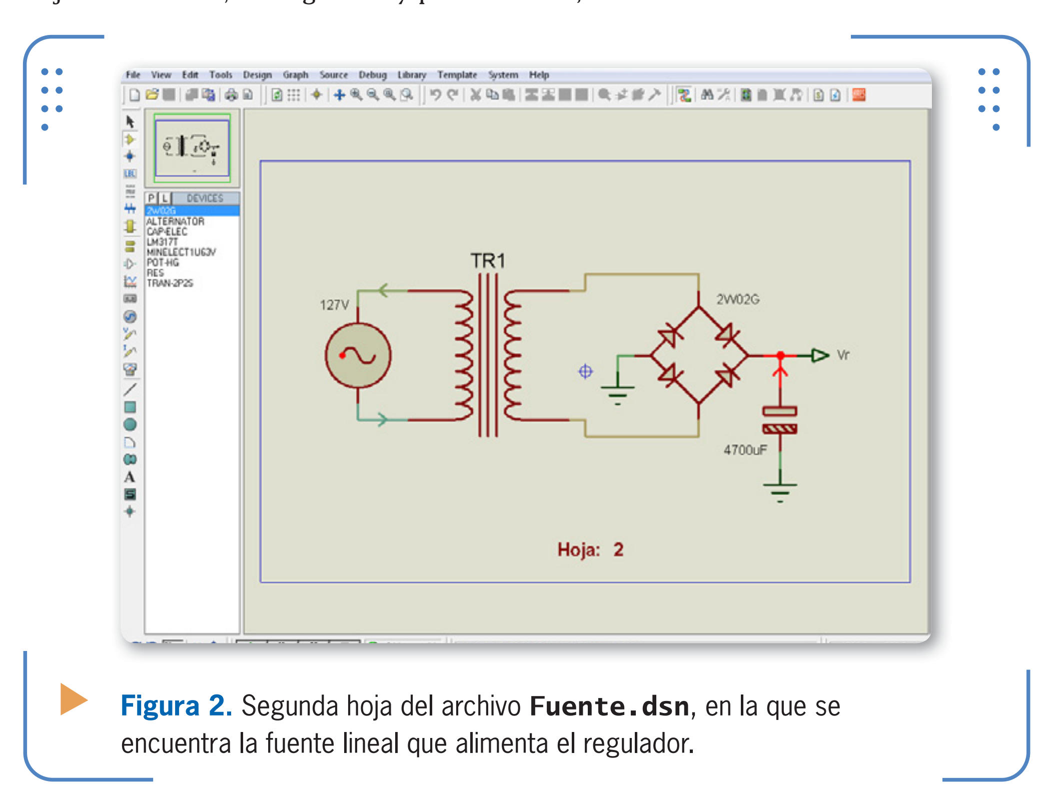


Figura 2. Segunda hoja del archivo **Fuente.dsn**, en la que se encuentra la fuente lineal que alimenta el regulador.

Es posible utilizar múltiples hojas para diagramas grandes divididos en partes o secciones, o para resaltar en la primera hoja el circuito principal. Si estamos dividiendo el diagrama en partes, la interconexión

entre hojas se dará a través de etiquetas de líneas de conexión. Recordemos que dos líneas (o buses) con el mismo nombre estarán interconectadas; el método más usual es a través de **terminales**. Tal como hemos hecho en el caso de la fuente, la terminal llamada **Vr** interconecta la salida de la fuente con la entrada del regulador.

Propiedades de las hojas

Si vamos al menú **Design** y elegimos **Edit Sheet Properties**, se abre una ventana donde podemos editar las propiedades de la hoja activa.

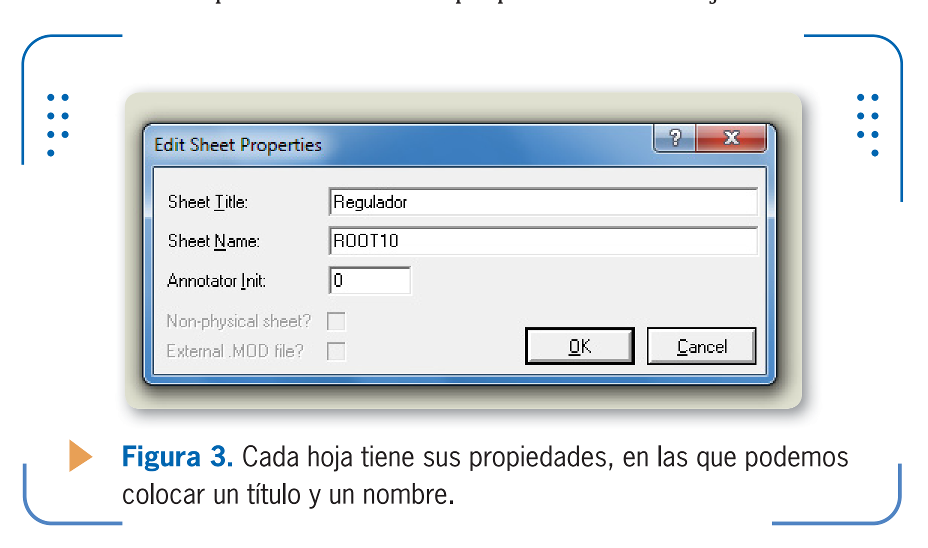


Figura 3. Cada hoja tiene sus propiedades, en las que podemos colocar un título y un nombre.

En el campo **Sheet title** ingresamos un título para la hoja, así damos una descripción de lo que contiene. En el ejemplo de la fuente, las hojas tienen el título **Regulador** y **Fuente**, respectivamente; este es el nombre que se muestra en la lista del menú **Design**. En el campo **Sheet Name** se especifica un nombre para la hoja, y este determinará la posición de la hoja en la lista. Por defecto, las hojas son nombradas como **ROOT10**, **ROOT20**, **ROOT30**. La palabra **root** significa **raíz**, por lo que cada hoja del diseño es una hoja raíz. Se numeran de diez en diez para permitir intercalar hojas nuevas entre ellas. Por ejemplo, si en el circuito de la fuente quisiéramos agregar una hoja entre la primera y la segunda, podríamos nombrar la nueva hoja como **ROOT15** y eso la colocará como la segunda hoja, en tanto que la **ROOT20** (Fuente) pasará a ser la tercera.



Subcircuitos

Otra forma de organizar o dividir los circuitos en un diagrama es mediante **subcircuitos**. Para entenderlo más claramente, podemos pensar en un diseño con subcircuitos como un diagrama de bloques, en el cual cada bloque representa un circuito particular. Por ejemplo, imaginemos un amplificador de audio, donde tendremos dos canales (derecho e izquierdo) y una fuente de alimentación.

Figura 4. Cada bloque que vemos en el diagrama representa un **subcircuito**, que forma parte del circuito completo.

Como podemos apreciar en la **Figura 4**, cada bloque del diagrama representa una parte del circuito total, y de esta forma hemos dividido el circuito completo en varios subcircuitos. Cada bloque o subcircuito contiene la parte que le corresponde.

La diferencia entre dividir el diseño en varias hojas o en subcircuitos radica en que, con los subcircuitos, todo el diseño puede mostrarse en una sola hoja (o en un solo diagrama).

El modo Subcircuito

Para crear un subcircuito, debemos entrar en modo **Subcircuito**. Para esto existe un botón en la barra de herramientas de **Modo** llamado **Subcircuit Mode**. Al presionarlo, ya podemos dibujar un bloque de subcircuito; simplemente, hacemos un clic en algún lugar vacío de la ventana de edición y arrastramos para formar un recuadro.

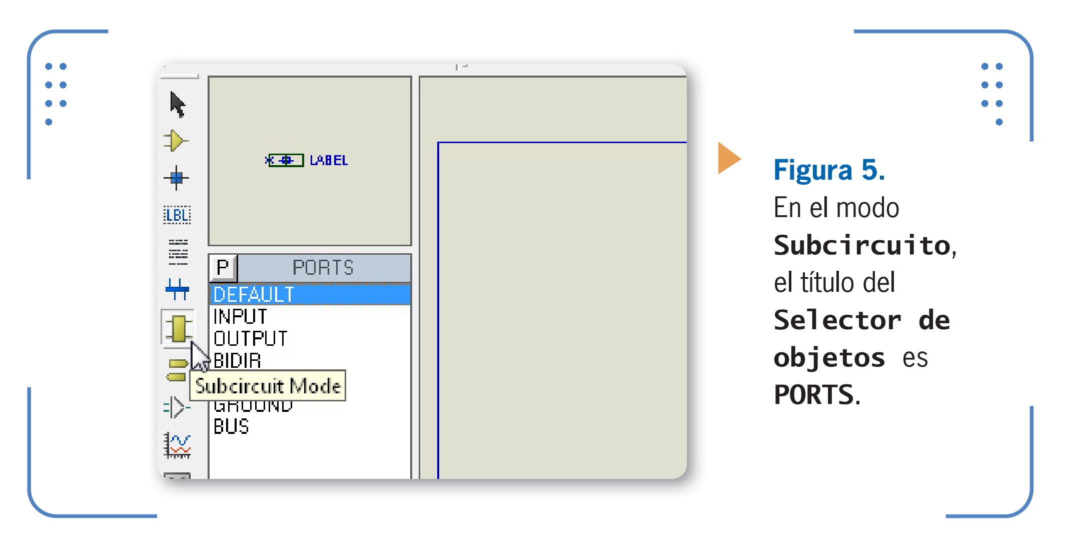


Figura 5. En el modo **Subcircuito**, el título del **Selector de objetos** es **PORTS**.

Al crear el recuadro, notaremos que la línea que lo delimita es gruesa y de color azul, esto indica que estamos trabajando con un subcircuito. Siempre que creemos uno nuevo, el nombre será por defecto **SUB?**. Para editarlo, accedemos a las propiedades del subcircuito, como en todo componente. Allí tenemos los campos **Name** y **Circuit**, que cambiaremos por los deseados. El campo **Name** define la etiqueta del subcircuito, y no podemos tener más de uno con el mismo nombre. En cambio, en el campo **Circuit** colocaremos una descripción, que puede ser igual para varios subcircuitos, sin ningún problema.



SUBCIRCUITOS Y CIRCUITOS INTEGRADOS

Al crear un subcircuito, podemos imaginar que estamos creando un circuito integrado, el cual tendrá un diagrama dentro que es funcional. Los puertos de subcircuito pueden representar los pines del CI; una vez construido, lo conectaremos en el diagrama como si fuera un circuito integrado más.

Puertos de subcircuito

Al entrar en modo **Subcircuito**, el título del **Selector de objetos** toma el nombre de **PORTS**, y en él se muestra la lista de **puertos de subcircuito**. La conexión desde o hacia un subcircuito se hará mediante terminales y puertos de subcircuito. Los puertos deben colocarse en los bordes derecho e izquierdo del bloque de subcircuito ya dibujado, y representan diferentes señales y sentidos.

PUERTOS

PUERTO	DESCRIPCIÓN
DEFAULT	Puerto de uso general.
INPUT	Para representar una entrada de señal hacia el subcircuito.
OUTPUT	Para representar una salida de señal de un subcircuito.
BIDIR	Representa un puerto bidireccional.
POWER	Puerto de alimentación.
GROUND	Puerto de tierra (GND).
BUS	Se usa para colocar un bus.

Tabla 1. Diferentes puertos de un subcircuito.

La mayoría de los puertos funcionan de manera idéntica, excepto el de bus, que debe conectarse a un pin de bus o ser manejado como un bus. Los demás solo se usan para representar diferentes tipos de señales o conexiones individuales, y se pueden utilizar según el gusto del usuario. Por ejemplo, una señal de salida puede ser representada con un puerto de salida, con uno de uso general o, incluso, con algún otro. Para agregar un puerto, debemos seleccionarlo en el **Selector de objetos** y llevar el cursor del mouse a un borde lateral de un subcircuito; este tomará la forma de un lápiz de color blanco con una cruz en la punta, y al hacer clic en ese lugar, el puerto quedará colocado. Como podemos notar, el procedimiento para agregar puertos en los diagramas de los circuitos es muy sencillo.

Hojas principales y subhojas

En cualquier diseño que contenga subcircuitos, la hoja donde estos se encuentran es considerada la hoja padre o principal (**parent sheet**), mientras que cada hoja de subcircuito es una hoja hija o subhoja (**child sheet**). Para acceder a una subhoja, hacemos un clic derecho sobre alguno de los bloques de subcircuito y en el menú contextual elegimos **Goto Child Sheet**. Con esto se abrirá la subhoja correspondiente. Si acabamos de crear ese subcircuito, estará vacía, y es aquí donde debemos dibujar o colocar el circuito deseado. Para regresar a la hoja principal, hacemos un clic derecho en un lugar vacío de la subhoja, y en el menú contextual elegimos **Exit To Parent Sheet**.

CADA HOJA DE SUBCIRCUITO ES CONSIDERADA UNA HOJA HIJA O SUBHOJA

Es posible combinar el uso de múltiples hojas y subcircuitos; por ejemplo, podemos tener un diseño con dos hojas y varios subcircuitos en cada una, incluso puede haber subcircuitos dentro de un subcircuito. Esto queda a gusto del diseñador y según la complejidad o extensión del circuito que vamos a dibujar.

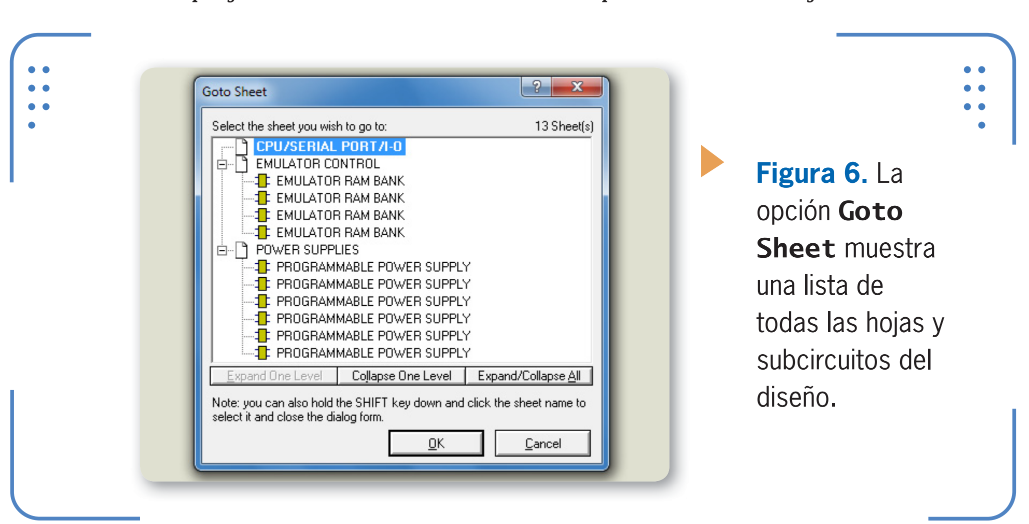


Figura 6. La opción **Goto Sheet** muestra una lista de todas las hojas y subcircuitos del diseño.

Si usamos la opción **Goto Sheet** del menú **Design** cuando tenemos múltiples hojas y subcircuitos en un diseño, en su ventana se mostrará una lista para explorar. Cada icono con una hoja blanca representa una hoja, y cada icono amarillo, un subcircuito (o subhoja).

Ejemplo

Como ejemplo utilizaremos otra vez la fuente variable que vimos antes dividida en varias hojas, pero ahora tomaremos la fuente y la colocaremos como un subcircuito. Usaremos el archivo **FuenteSub.dsn**.

Figura 7. La fuente variable que antes teníamos en varias horas, en este ejemplo, la tenemos con un subcircuito.

Ahora, en lugar de ubicar la fuente lineal en otra hoja, la hemos colocado como un subcircuito llamado **SUB 01**. Para ver el diagrama de la fuente, debemos acceder a la subhoja con la opción **Goto Child Sheet**.





LAS HOJAS DE DATOS

Normalmente debemos consultar las hojas de datos de los componentes para poder dibujarlos en el módulo ISIS. En ellas podremos observar el número de pines, el nombre, la numeración y la función de cada uno. En Internet existen diversos lugares para poder descargar gratuitamente las hojas de datos de diferentes componentes, un ejemplo es: **www.datasheetcatalog.com**.

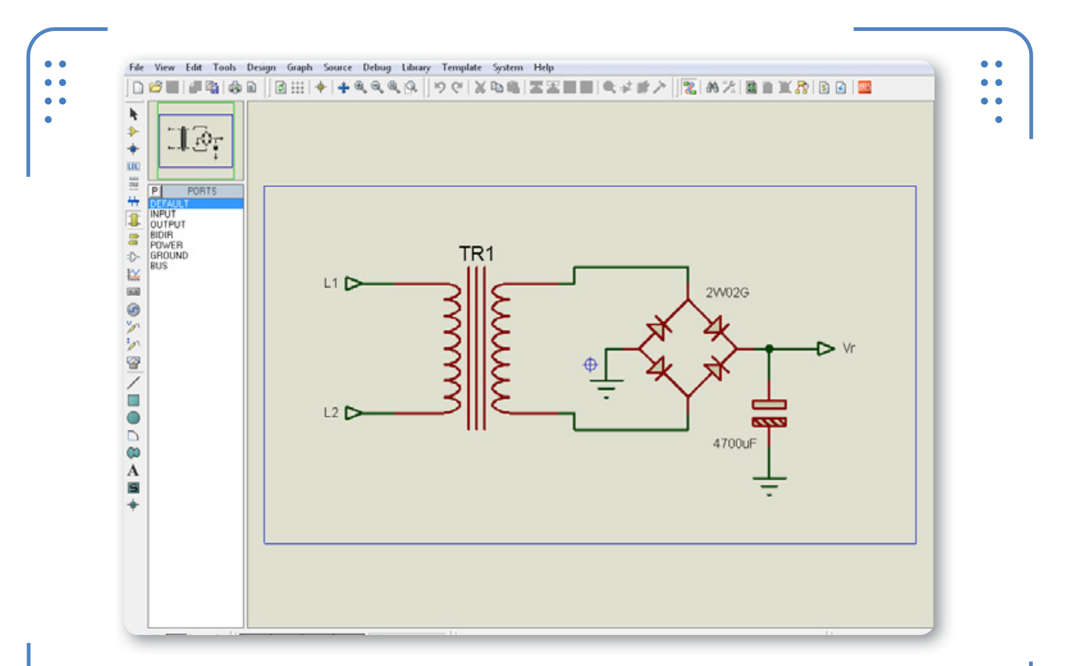


Figura 8. En este caso, la subhoja llamada **SUB 01** es la que contiene el diagrama de la fuente de poder.

Notaremos que las terminales **L1** y **L2** se conectan con los **puertos** respectivos en el subcircuito, son la entrada de la línea; y la terminal **Vr** se conecta con su respectivo **puerto**, es la salida de la fuente hacia el regulador. En este ejemplo, tenemos una sola hoja que contiene todos los circuitos y subcircuitos de la fuente.

La división de los diagramas en hojas o subcircuitos nos permite trabajar de una manera más ordenada y profesional.



COMPONENTES PERSONALIZADOS

Prácticamente cualquier componente existente en las librerías del módulo ISIS puede ser modificado usando el comando **Decompose** para cambiar su aspecto, el tamaño, la distribución de los pines, etcétera. Una buena práctica es ir creando poco a poco una colección de componentes personalizados para poder usarlos cada vez que los necesitemos. Esta posibildad de adaptar los componentes es otra de las características interesantes que tiene Proteus.



Creación y modificación de componentes

ISIS nos permite modificar los componentes disponibles o crear nuevos. El símbolo de un componente es, básicamente, una parte gráfica más los pines que lo forman. Para la parte gráfica, podemos usar las herramientas de dibujo en 2D; para agregar pines, debemos entrar en **modo de pines de dispositivo** con el botón correspondiente de la barra de herramientas de **Modo**.

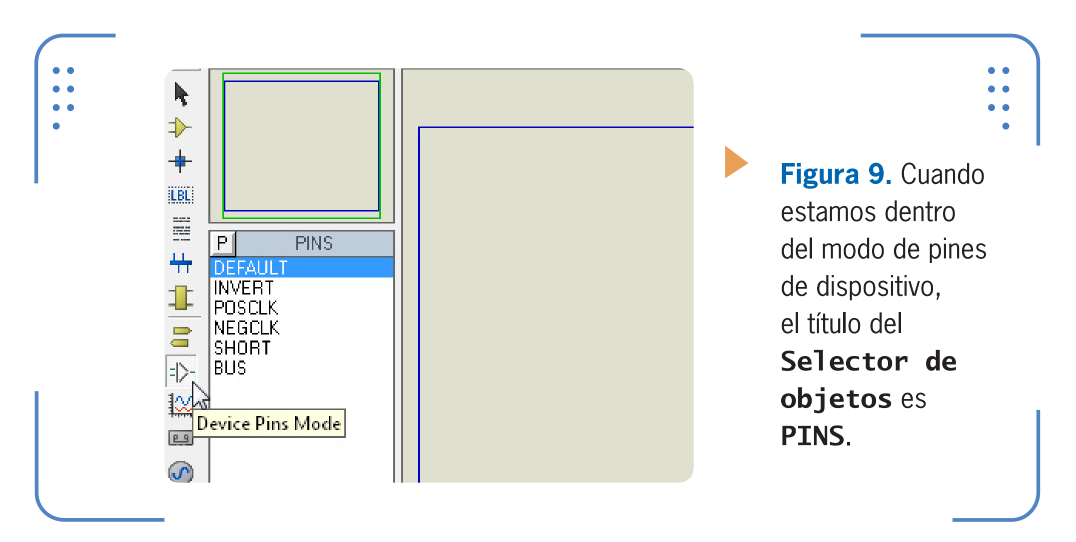


Figura 9. Cuando estamos dentro del modo de pines de dispositivo, el título del **Selector de objetos** es **PINS**.

Al presionar el botón **Device Pins Mode**, en el **Selector de objetos** aparecerá la lista de pines disponibles. **DEFAULT** es el más usado, ya que es un pin genérico. También encontramos pines inversores, de reloj positivo y negativo, un pin corto y uno de bus, entre algunos



PERSONALIZAR LAS LISTAS DE MATERIALES

Es posible cambiar la forma en que se presenta la lista de materiales. Para hacerlo, vamos al menú **System/ Set BOM Scripts...**. Se abrirá una ventana donde podemos personalizar el texto, las columnas, los datos que contiene cada columna, y otros detalles referidos a cómo se muestra una lista de materiales al invocarla.

otros. Al seleccionar un pin de la lista, veremos que aparece en la ventana de vista previa con un número **99**, el cual representa el número del pin; la cruz en la punta indica el lugar donde se conectará el pin a una línea de conexión; y el cuadro indica la base, es decir, el lugar donde el pin se unirá al componente.

Crear un circuito integrado DS1624

Veamos un ejemplo paso a paso de cómo crear un dispositivo o símbolo nuevo. En este caso, tomaremos un circuito integrado **DS1624**, que no se encuentra en Proteus. El primer paso es, por supuesto, saber qué tipo de circuito integrado es y cuántos pines tiene.

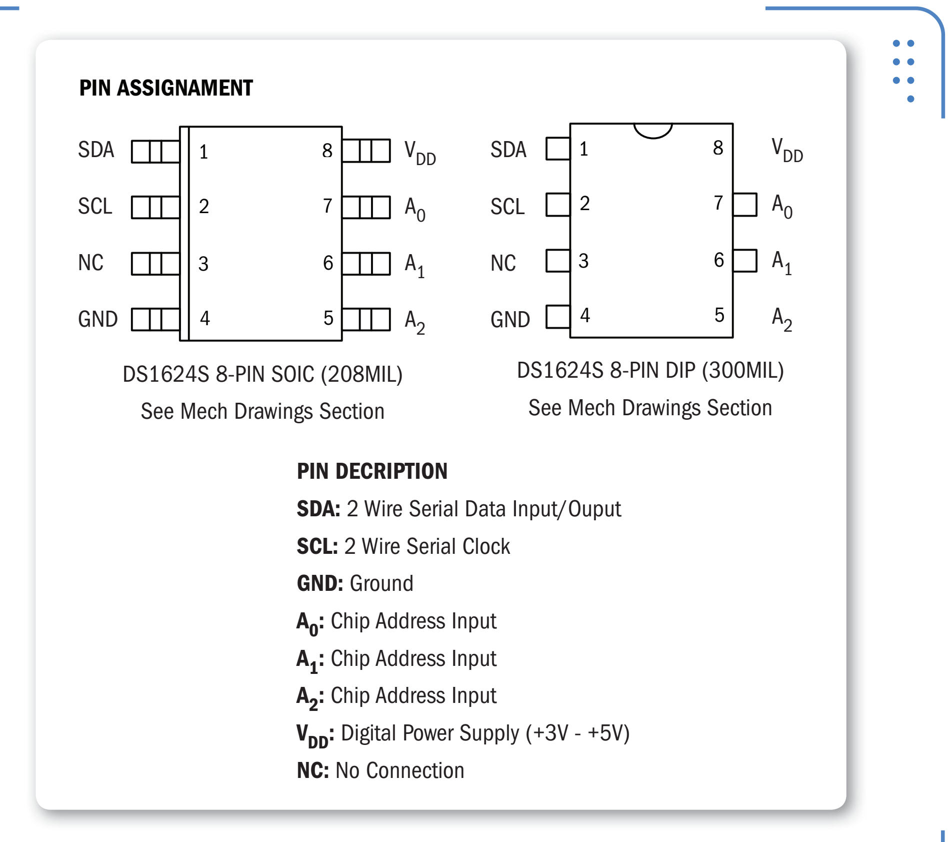


Figura 10. Tomaremos una hoja de datos del circuito integrado **DS1624** para crear el dispositivo en base a ella.

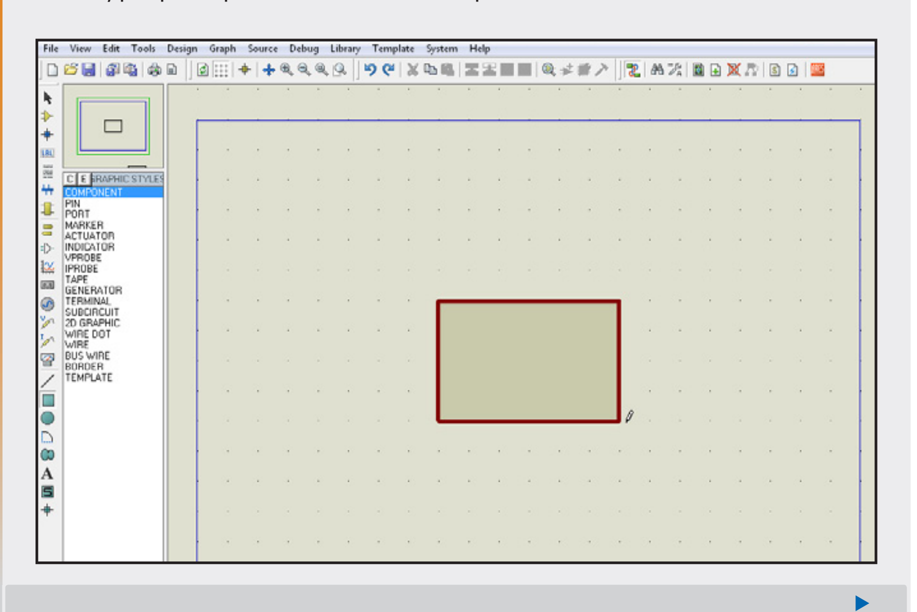
Para esto podemos buscar en Internet y consultar la hoja de datos del componente que queremos construir. ISIS permite crear nuevos componentes en la misma ventana de edición; incluso es posible hacerlo mientras estamos trabajando en un diseño, simplemente tenemos que usar un espacio que se encuentre vacío.

▼ PASO A PASO: DIBUJAR UN NUEVO COMPONENTE





Haga clic en el botón `2D Graphics Box Mode` en la barra de herramientas de Modo, y en el `Selector de objetos` elija el estilo **`COMPONENT`**. Dibuje un rectángulo, que será el cuerpo del circuito integrado. No se preocupe mucho por el tamaño, porque se puede redimensionar después.





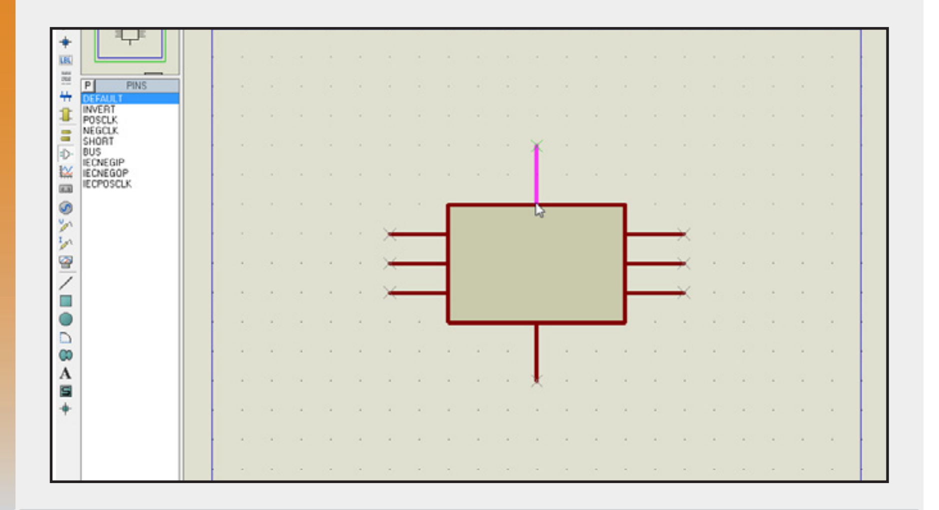


COMPONENTES NUEVOS Y SIMULACIÓN

Cuando creamos un componente nuevo, este solo se usará para el dibujo de diagramas; no puede simularse, ya que no cuenta con un modelo de simulación. En algunos casos, es posible conseguir el modelo SPICE de un componente, pero tendremos que realizar configuraciones avanzadas para lograr un nuevo componente que pueda ser simulable.

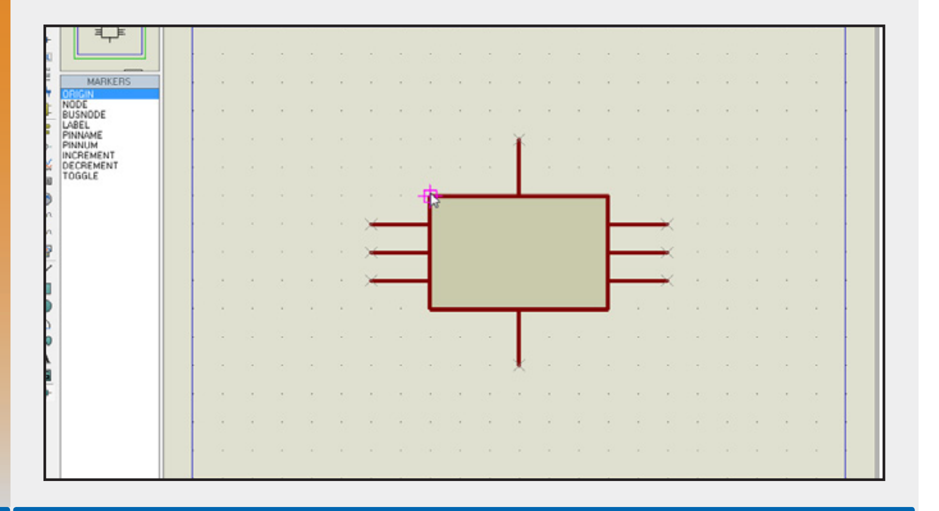
02

Haga clic en el botón `Device Pins Mode` y seleccione el pin DEFAULT. Coloque los pines en el lugar deseado, y use los comandos de rotación o ref exión para ubicar los pines a los costados o arriba y abajo.



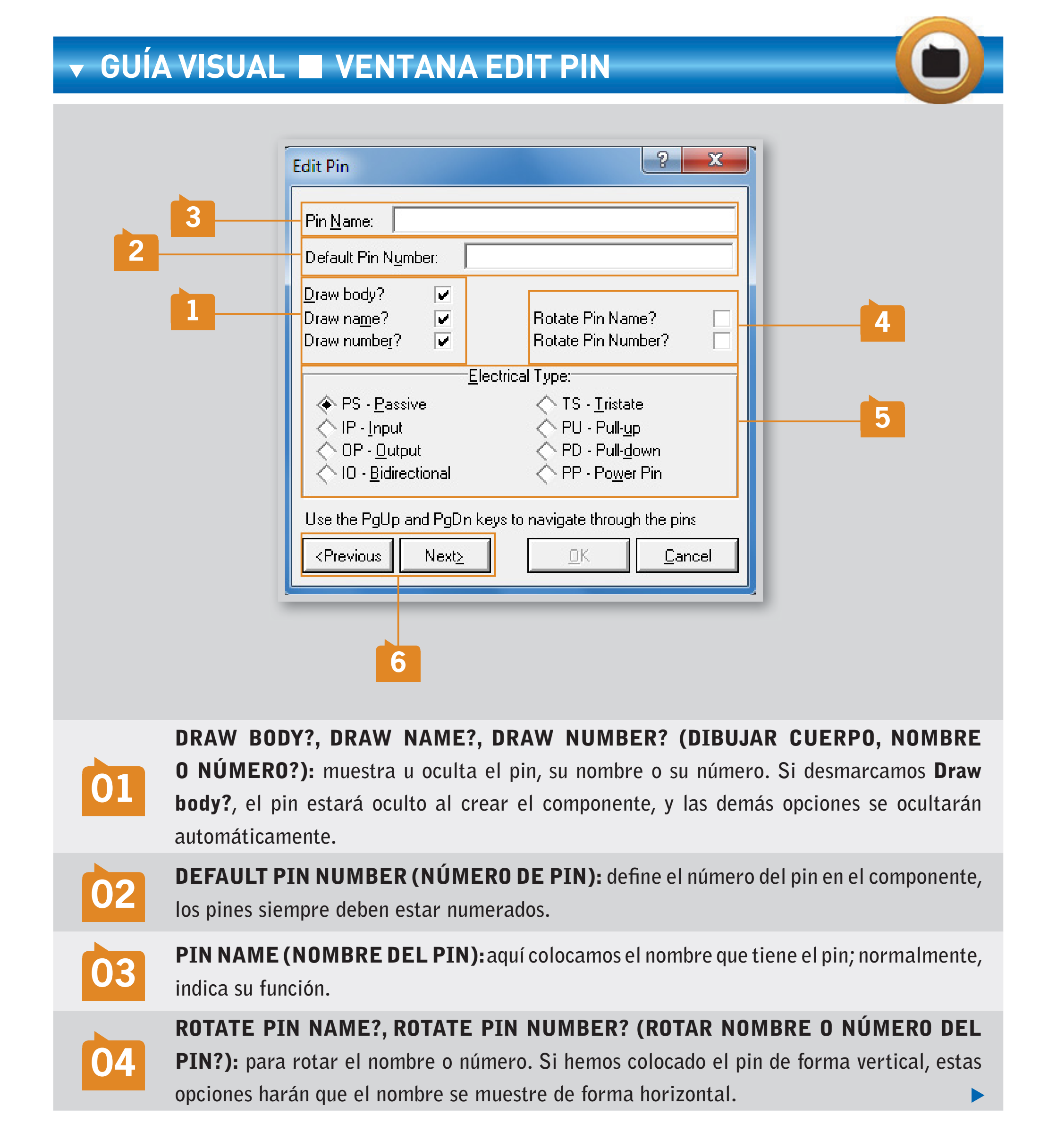
03

Haga clic en el botón `2D Graphics Markers Mode` y coloque un marcador en la esquina superior izquierda del cuerpo del componente. Este indicará el lugar donde aparecerá centrado el nuevo componente.



Hasta aquí tenemos el dibujo básico del nuevo componente. Hemos colocado los pines de alimentación arriba y abajo, respectivamente, porque en este caso estarán ocultos. Ahora debemos numerar y nombrar los pines, para lo cual accedemos a las propiedades de cada pin. Hacemos un doble clic en cualquiera de ellos, para abrir la ventana **Edit Pin**. En la siguiente **Guía visual** conoceremos cada uno de sus elementos.

GUÍA VISUAL ■ VENTANA EDIT PIN



01 **DRAW BODY?, DRAW NAME?, DRAW NUMBER? (DIBUJAR CUERPO, NOMBRE O NÚMERO?):** muestra u oculta el pin, su nombre o su número. Si desmarcamos **Draw body?**, el pin estará oculto al crear el componente, y las demás opciones se ocultarán automáticamente.

02 **DEFAULT PIN NUMBER (NÚMERO DE PIN):** define el número del pin en el componente, los pines siempre deben estar numerados.

03 **PIN NAME (NOMBRE DEL PIN):** aquí colocamos el nombre que tiene el pin; normalmente, indica su función.

04 **ROTATE PIN NAME?, ROTATE PIN NUMBER? (ROTAR NOMBRE O NÚMERO DEL PIN?):** para rotar el nombre o número. Si hemos colocado el pin de forma vertical, estas opciones harán que el nombre se muestre de forma horizontal.



ELECTRICAL TYPE (TIPO DE PIN): aquí elegimos el tipo eléctrico del pin, por ejemplo, si es salida, entrada, bidireccional, pin de alimentación, etcétera.



PREVIOUS, NEXT (ANTERIOR, SIGUIENTE): permiten navegar entre los pines del componente. Al presionarlos, se pasará al pin anterior o siguiente, de modo que podemos editar todos los pines desde esta ventana. El pin seleccionado se mostrará en rojo en el componente.

Al hacer doble clic sobre un pin, podemos editar las propiedades, darles un nombre y un número. Es posible colocar los pines en el orden deseado, siempre que la numeración esté completa.

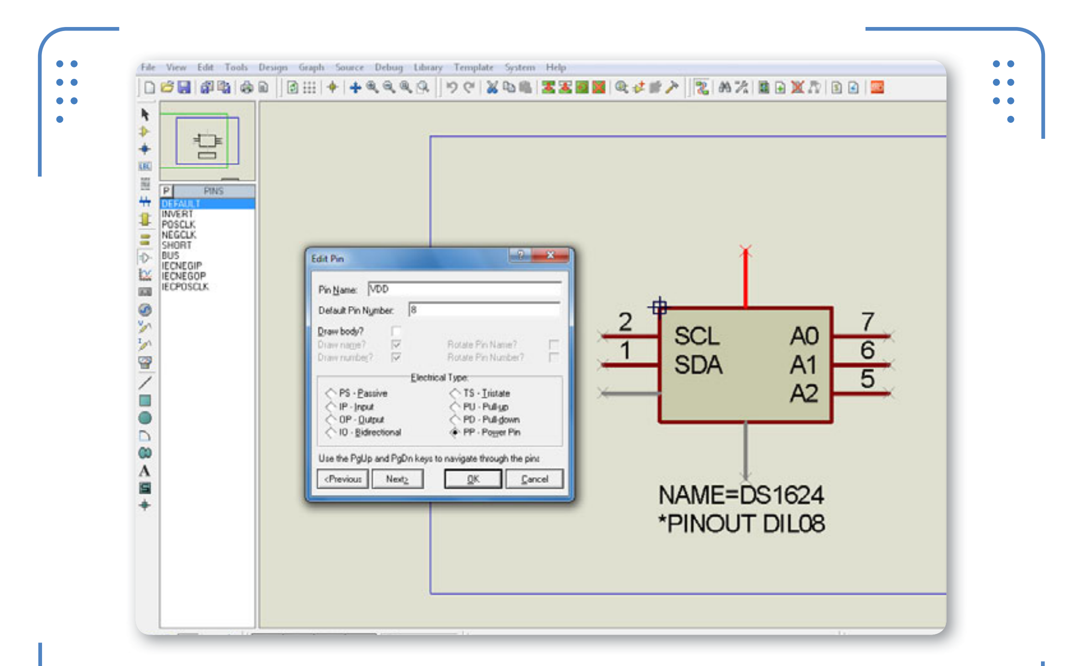


Figura 11. A través de la ventana **Edit Pin** es posible navegar por todos los pines que posee el componente.

Los pines 1 y 2 se configuran como **bidireccionales**; el pin 3, como **pasivo** (no tiene conexión y estará oculto); el pin superior **VDD** y el inferior **GND**, como pines de alimentación y estarán ocultos; y los pines 5, 6 y 7 son **entradas**. Los pines ocultos se muestran en gris.

Ya estamos listos para crear el nuevo componente, para lo cual seleccionamos todos los elementos que lo integran. Hacemos un recuadro alrededor de los elementos, arrastrando el cursor del mouse, y dando un clic para encerrar todo, así quedarán seleccionados. Luego, pulsamos el comando **Make device**, ya sea en la barra de herramientas de **Edición**, en el menú contextual al hacer un clic derecho o en el menú **Library**.

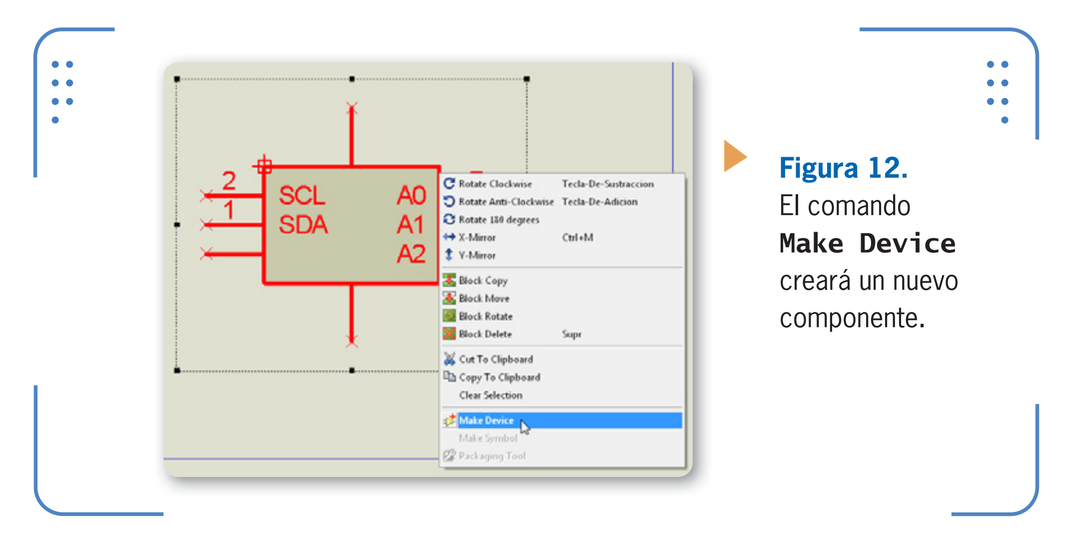


Figura 12. El comando **Make Device** creará un nuevo componente.

Al hacerlo, aparecerá una ventana llamada **Make Device**, donde configuraremos algunas opciones para el nuevo circuito integrado. Esta ventana tiene el título **Device Properties**.

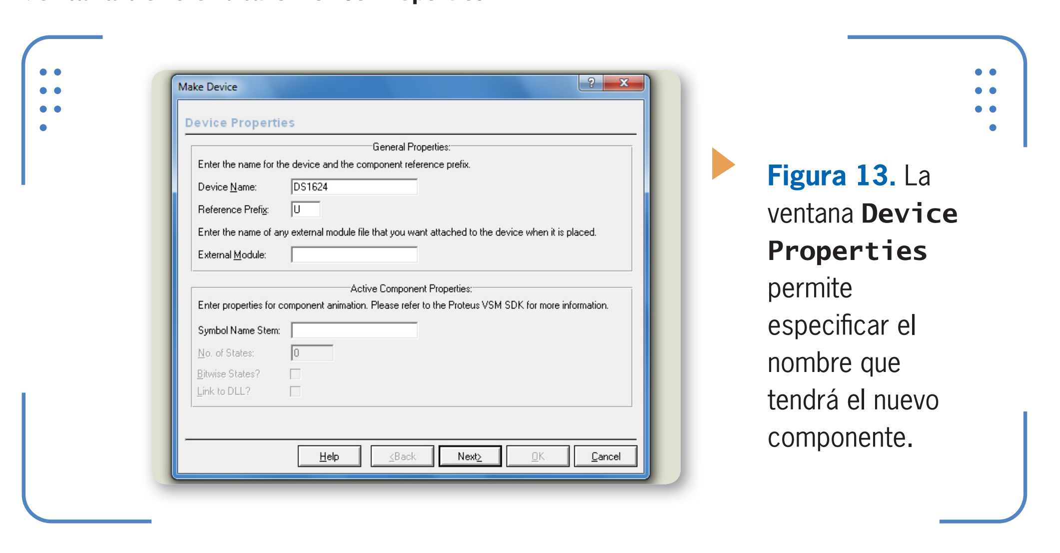


Figura 13. La ventana **Device Properties** permite especificar el nombre que tendrá el nuevo componente.

En esta primera ventana, debemos asignarle un nombre al dispositivo. En este caso, ingresamos **DS1624** en el campo **Device Name**. En el campo **Reference Prefix**, colocaremos **U** como en todos los circuitos integrados. Al presionar sobre el botón **Next**, aparecerá una segunda ventana con el título **Packagings**.

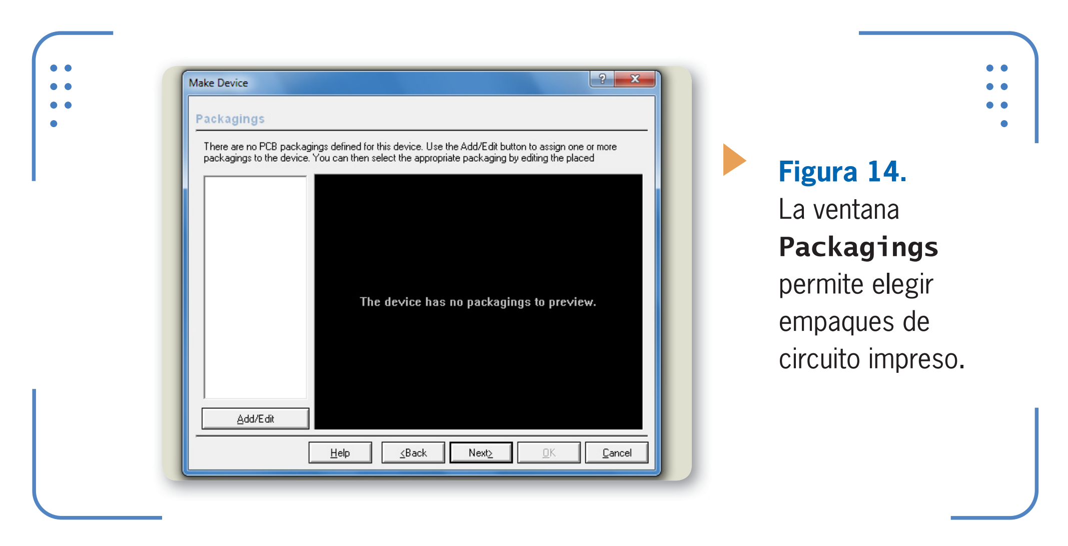


Figura 14. La ventana **Packagings** permite elegir empaques de circuito impreso.

En esta ventana, podemos elegir un **empaque** para el componente. En caso de no necesitar hacerlo, simplemente presionamos el botón **Next** para continuar. Si queremos asignar un empaque, pulsamos el botón **Add/Edit**, que abrirá la ventana **Package Device**.

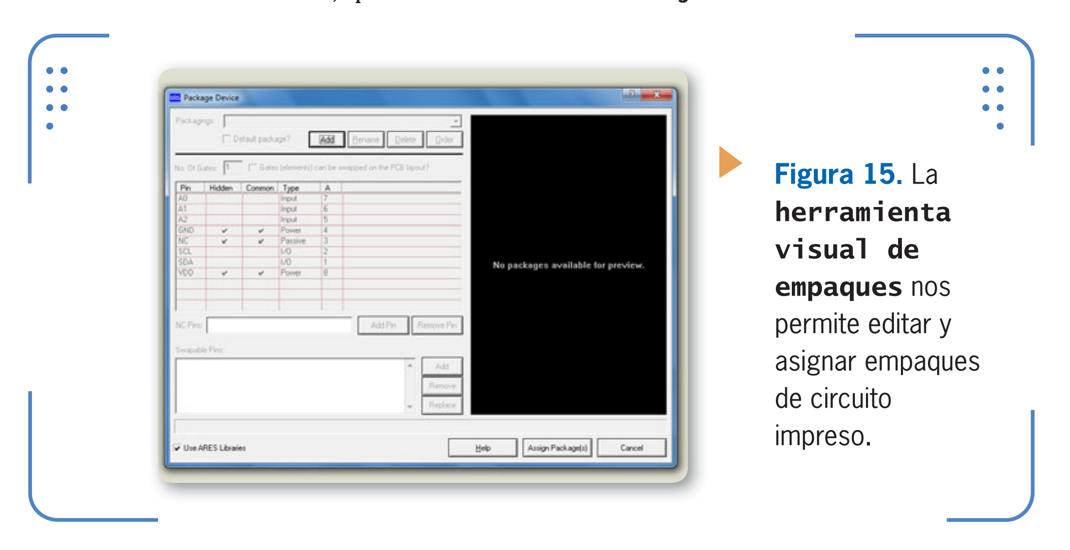


Figura 15. La **herramienta visual de empaques** nos permite editar y asignar empaques de circuito impreso.

En esta nueva ventana veremos la lista de todos los pines, con su nombre y número. Para continuar debemos presionar el botón **Add** y se abrirá la ventana **Pick Packages**, donde podremos encontrar el empaque adecuado para nuestro circuito integrado.

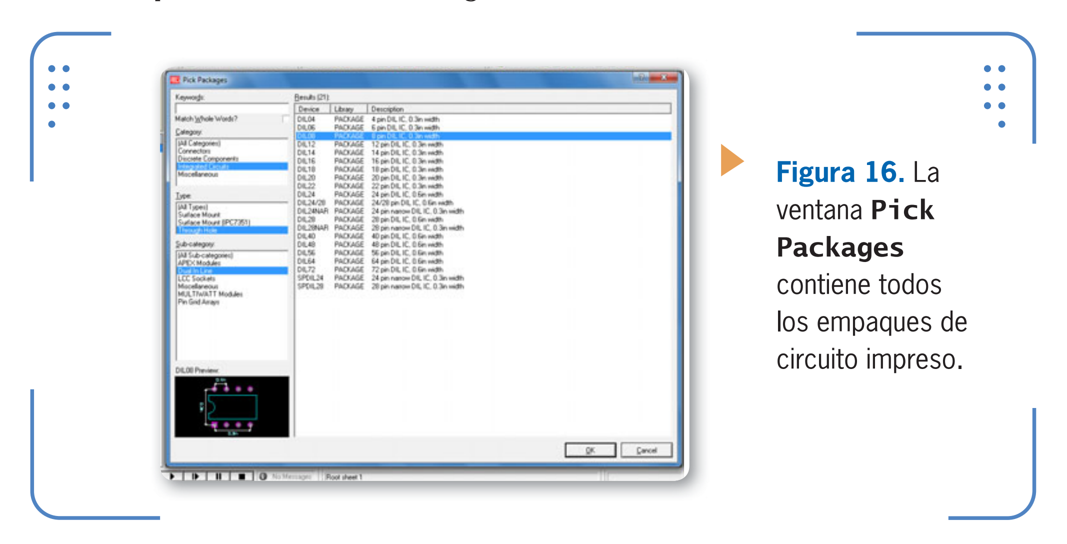


Figura 16. La ventana **Pick Packages** contiene todos los empaques de circuito impreso.

En el campo **Category:** seleccionamos **Integrated Circuits**; en el campo **Type:** elegimos **Through Hole**; y en **Sub-category:** pulsamos **Dual In Line**. Finalmente, en **Results:**, elegimos el empaque **DIL08**. Al hacer clic en **OK**, regresamos a la ventana **Package Device** con el empaque asignado, donde podemos modificar los pines e incluso agregar otros, si lo deseamos.

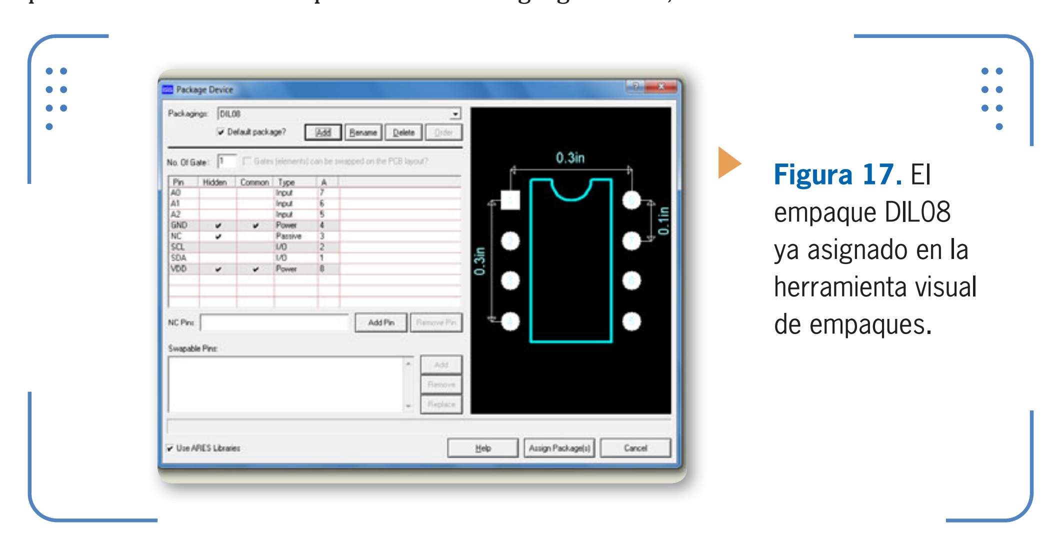


Figura 17. El empaque DIL08 ya asignado en la herramienta visual de empaques.

Hacemos clic en **Assign Package(s)** y el empaque será asignado al componente; debemos saber que es posible asignar más de un empaque a un componente. En la ventana **Packagings**, presionamos **Next**, para acceder a **Component Properties & Definitions**, donde definimos los parámetros avanzados del componente.

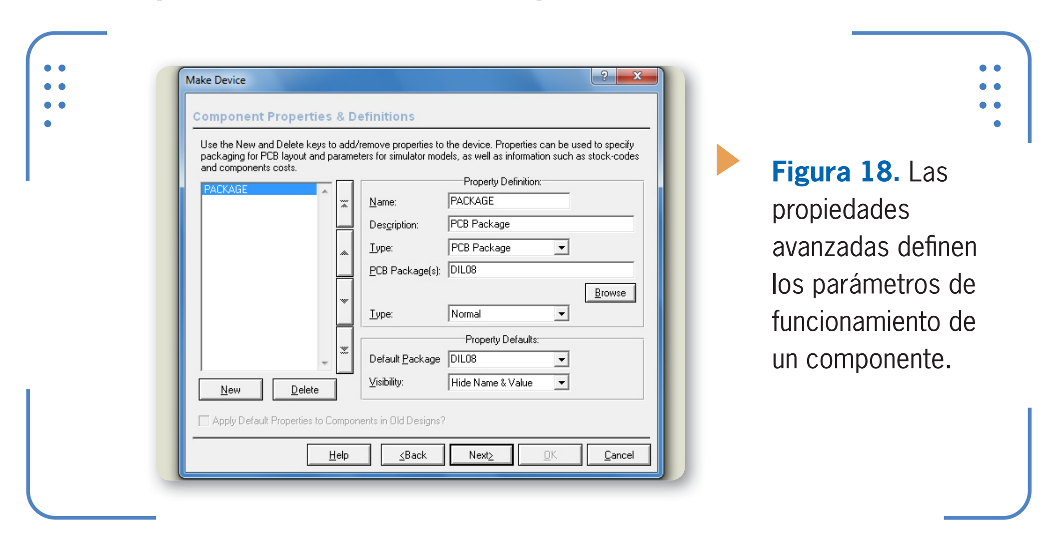


Figura 18. Las propiedades avanzadas definen los parámetros de funcionamiento de un componente.

Por el momento no modificaremos nada en esta ventana, simplemente, debemos hacer un clic en el botón **Next** para continuar con el proceso de configuración. Se abrirá la ventana denominada **Device Data Sheet & Help File**.

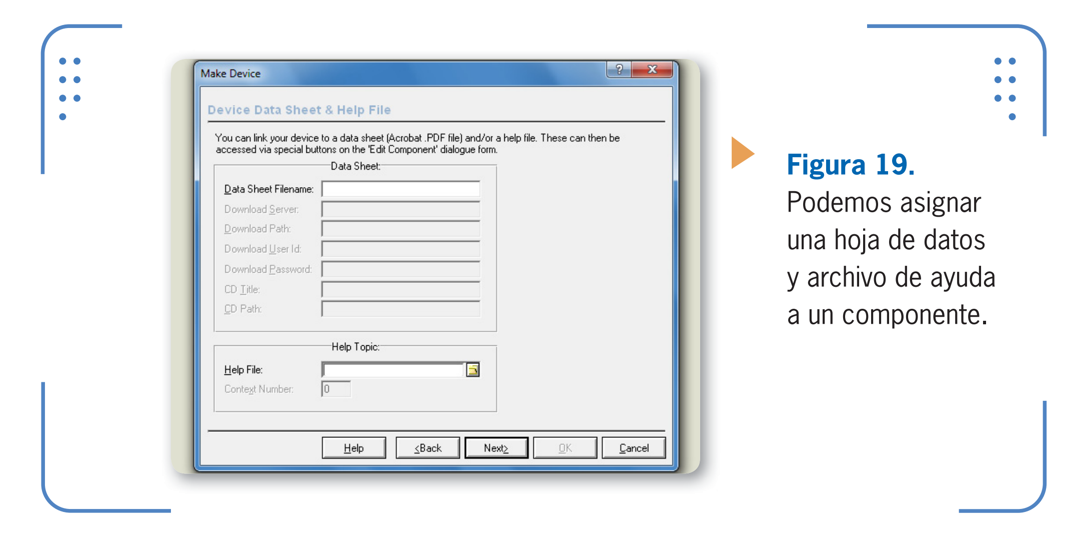


Figura 19. Podemos asignar una hoja de datos y archivo de ayuda a un componente.

Aquí podemos indicar el lugar donde se encuentra la hoja de datos y el archivo de ayuda para el nuevo componente. Como por el momento no nos interesa esto, pulsamos **Next**, para acceder a la última ventana, **Indexing and Library Selection**.

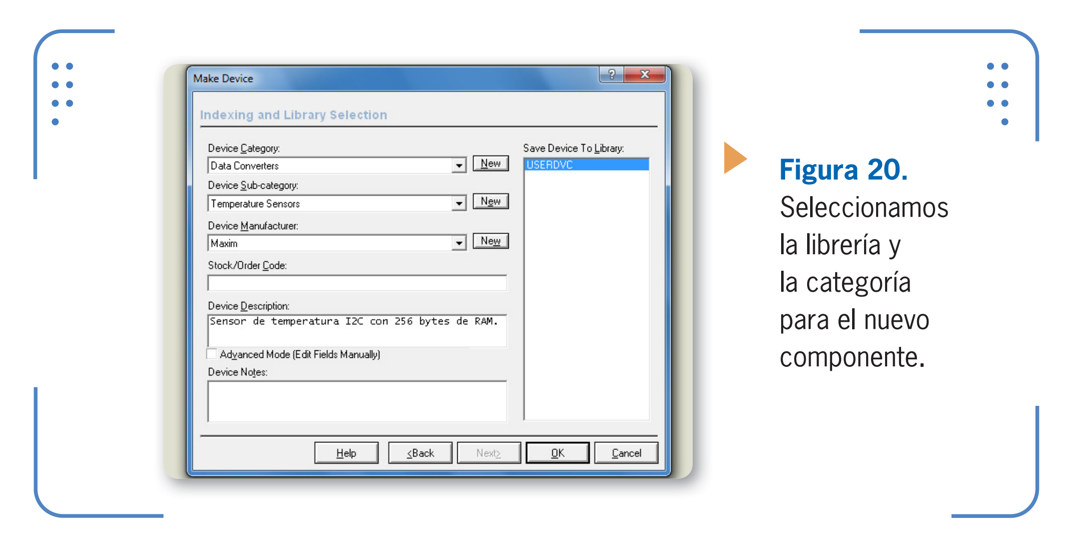


Figura 20. Seleccionamos la librería y la categoría para el nuevo componente.

En esta ventana elegimos la categoría a la que el nuevo componente será asignado y la librería donde se guardará. En **Device Category**, seleccionamos **Data Converters**, que es donde se encuentran los dispositivos similares a este. En el campo **Device Description** escribimos una breve descripción del componente. En la sección **Save Device To Library** debemos elegir la librería donde se guardará este nuevo componente; en nuestro caso, solo tenemos la librería **USERDVC**, que es la librería de usuario. Por lo tanto, la seleccionamos.

UNA VEZ GUARDADO EL COMPONENTE, APARECERÁ EN EL SELECTOR DE OBJETOS



Al hacer clic en **OK**, el nuevo componente quedará creado y guardado. Además, su nombre aparecerá en el **Selector de objetos** y ya podremos utilizarlo en nuestros diagramas. También será posible seleccionarlo o buscarlo en la ventana **Pick Devices**, como cualquier otro componente. A continuación, borramos los elementos que usamos para crearlo, ya que ahora carecen de utilidad. El proceso de creación de un nuevo componente es sumamente sencillo, y nos permite ampliar las posibilidades de diseño de los circuitos.

Modificar un componente existente

Podemos modificar un componente existente para cambiar algún aspecto, como su tamaño o la distribución de los pines. Simplemente debemos elegir el componente, colocarlo en la ventana de edición y, luego, seleccionar **Decompose** en el menú contextual, en la barra de herramientas de edición o en el menú **Library**. Al hacer esto, el componente será dividido en sus partes esenciales, tanto las gráficas como los pines. A partir de aquí, podemos hacer todas las modificaciones y después proceder de la misma forma en la que creamos un componente, con el comando **Make Device**.

Un teclado personalizado

Supongamos que deseamos probar el circuito integrado **MM74C922**, que es un decodificador de teclado. Podemos armar un circuito agregando un teclado, pero, en este caso, los que contiene ISIS no son exactamente como los que necesitamos.

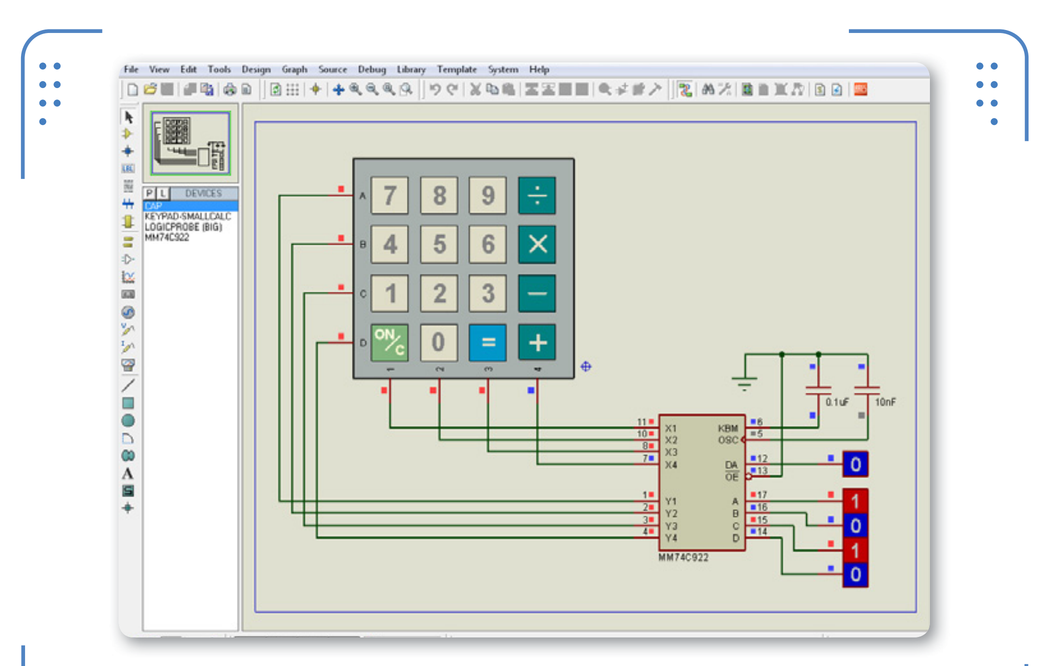


Figura 21. Tenemos un circuito con un teclado de la librería **ACTIVE**, que no es exactamente lo que buscamos.

El teclado más adecuado podría ser el que está diseñado para una calculadora (**KEYPAD-SMALLCALC**), pero como no se ajusta exactamente a lo que buscamos, vamos a modificarlo para poder adaptarlo al dispositivo que necesitamos.

Figura 22. Mediante la opción **Decompose**, dividimos el teclado en las diferentes partes que lo integran para modificarlo.

Como explicamos anteriormente, seleccionamos el teclado y elegimos el comando **Decompose** para modificarlo. A continuación, cambiamos las teclas por las numeradas del **0** al **15**, cambiamos la posición de los pines (esto es opcional), seleccionamos todo el nuevo



¿PARA QUÉ ASIGNAR EMPAQUES?

El hecho de asignar un empaque a un nuevo componente solo tiene utilidad real si estamos planeando diseñar circuitos impresos en ARES. Si no lo haremos, podemos omitir este paso. Es posible asignar el empaque más adelante, editando el componente ya creado o asignarlo manualmente al diseñar un PCB.

teclado e invocamos el comando **Make Device**. Al hacer esto, accederemos a las ventanas que nos permiten crear el nuevo componente. En la primera (**Device Properties**) solo cambiaremos el nombre del teclado en el campo **Device Name**, por ejemplo, **MY KEYPAD**. Esto servirá para no sobrescribir el componente original y conservarlo, así crearemos un teclado totalmente nuevo.

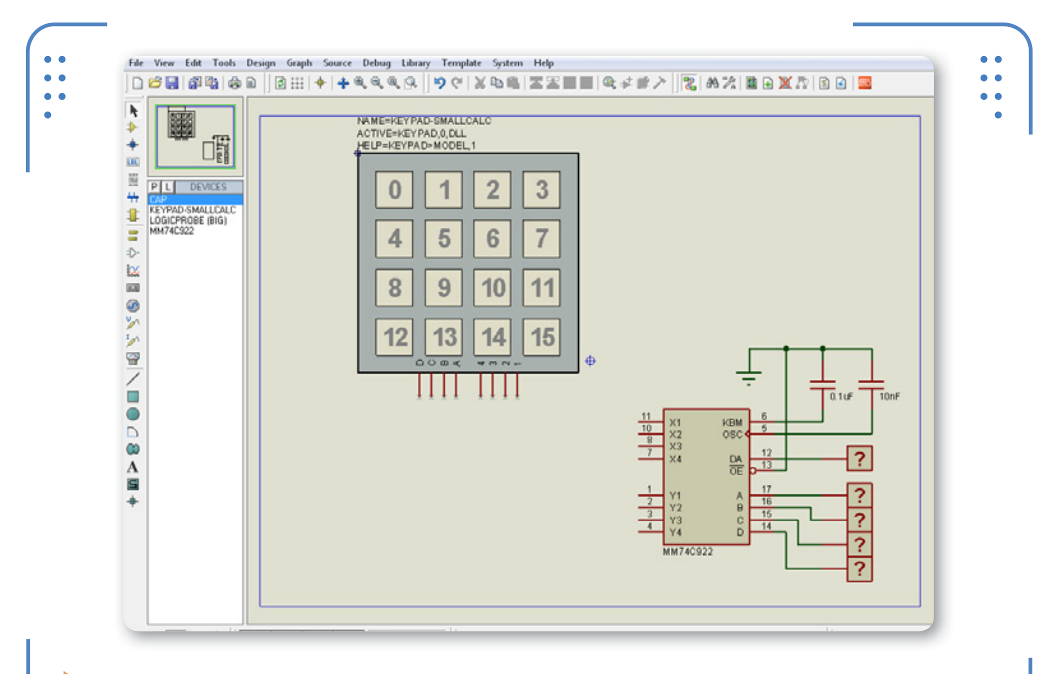


Figura 23. Tenemos el teclado ya modificado a nuestro gusto y listo para usar en la creación de un nuevo componente.

AL CONSERVAR LAS PROPIEDADES, PODEMOS SIMULAR EL CIRCUITO CON EL NUEVO TECLADO



Luego, hacemos clic en el botón **Next** hasta la última ventana en donde podemos cambiar la descripción, si lo deseamos. Al terminar y presionar el botón **OK**, tendremos nuestro teclado. Borramos los elementos que sirvieron para crearlo y colocamos el nuevo teclado personalizado, hacemos las conexiones correspondientes y, como no modificamos las propiedades ni los parámetros, podemos simular el circuito perfectamente con este nuevo teclado.

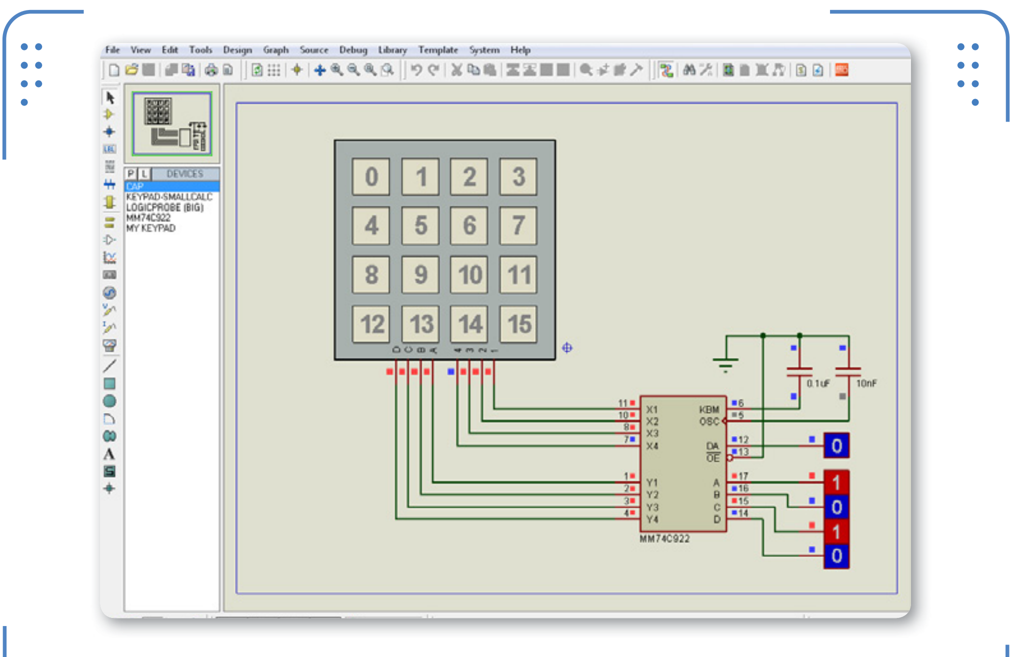


Figura 24. Ya está todo listo: podemos ver el circuito con nuestro nuevo teclado creado y funcionando.



Propiedades y notas del diseño

En el menú **Design** hay dos opciones para agregar notas en el diseño. La primera es **Edit Design Properties**; al seleccionarla, se abrirá una ventana con ese título. En la **Guía visual** conoceremos sus elementos.

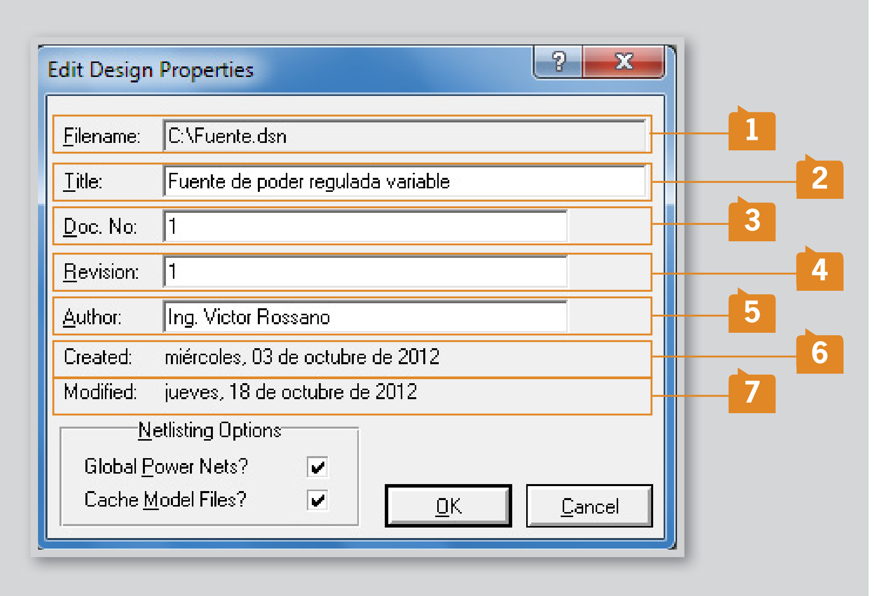


MODIFICAR COMPONENTES SIMULABLES

Al modificar un componente existente y que es simulable, podemos mantener la posibilidad de simulación si conservamos la cantidad, numeración o función de los pines. Cuando creamos el componente en la ventana **Component Properties & Definitions**, no debemos modificar sus parámetros.

GUÍA VISUAL ■ PROPIEDADES DEL DISEÑO





01 **FILENAME (NOMBRE DE ARCHIVO):** muestra el nombre y la ruta del archivo, es solo informativo y no puede modificarse.

02 **TITLE (TÍTULO):** aquí podemos colocar un título para el diseño, es independiente del nombre del archivo.

03 **DOC. NO (NÚMERO DE DOCUMENTO):** para ingresar el número del documento o archivo.

04 **REVISION (REVISIÓN):** para especificar un número de revisión del diseño.

05 **AUTHOR (AUTOR):** para indicar el nombre del autor del diseño.

06 **CREATED (CREADO):** muestra la fecha en que el archivo fue creado.

07 **MODIFIED (MODIFICADO):** informa la fecha en que el archivo fue modificado por última vez.

Estos datos pueden ser de utilidad para llevar un control del archivo, sobre todo, si estamos trabajando con circuitos que requieran de pruebas y revisiones de varias personas; de ahí la utilidad del número de revisión. Además, algunos datos de esta ventana aparecerán en otros sitios, como en **reportes** generados por ISIS y en los cuadros de datos en los diseños que los tengan.

La segunda opción que encontramos en el menú **Design** es **Edit Design Notes**; al seleccionarla, se abrirá una ventana con el mismo nombre, en donde podemos colocar notas acerca del diseño.

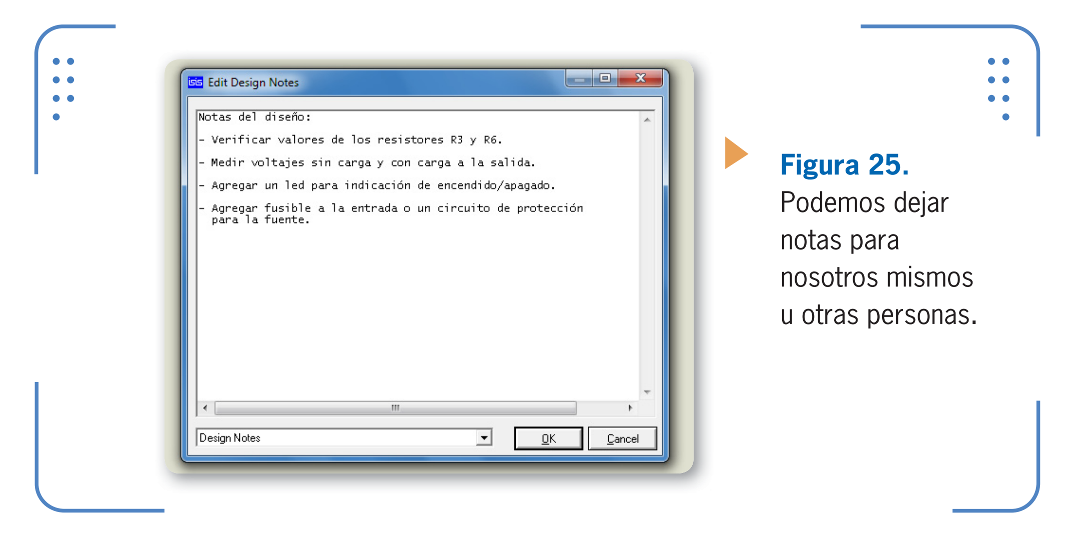


Figura 25. Podemos dejar notas para nosotros mismos u otras personas.

En la parte inferior tenemos una lista desplegable con dos opciones: **Design notes** y **Procurement notes**. La primera sirve, simplemente, para dejar comentarios sobre del diseño, por ejemplo, si necesita alguna modificación o se requiere revisar una parte específica de él. La segunda se puede traducir como **notas de compras**, y es útil para indicar si hace falta comprar algunos componentes o elementos para armar el diseño físicamente, o cualquier aclaración relacionada con la compra o adquisición de componentes. Estas notas pueden ser usadas libremente a gusto de cada uno y adquieren importancia, en especial, cuando un diseño es trabajado por diferentes personas.

CON LA OPCIÓN EDIT DESIGN NOTES PODREMOS AGREGAR NOTAS ACERCA DEL DISEÑO





Generación de reportes

ISIS puede generar diferentes reportes que permiten analizar un diseño, localizar posibles errores o generar listas de compras. A continuación, veremos algunos de ellos.

Lista de materiales (BOM)

ISIS puede generar de forma rápida y fácil una lista de los componentes y materiales usados en un circuito para saber cuántos son o comprarlos. Esta es la **lista de materiales**, también conocida como **BOM** (Bill Of Materials) según su abreviatura en inglés. Para generar la lista, debemos ir al menú **Tools** y luego a **Bill of Materials**. Hay cuatro posibilidades diferentes: HTML, ASCII, CVS compacto o CVS completo.

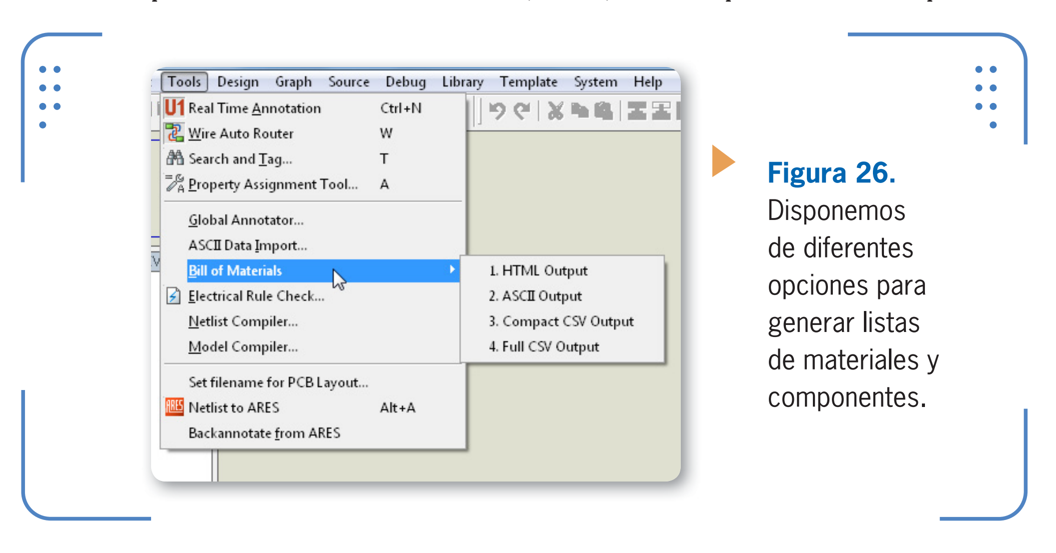
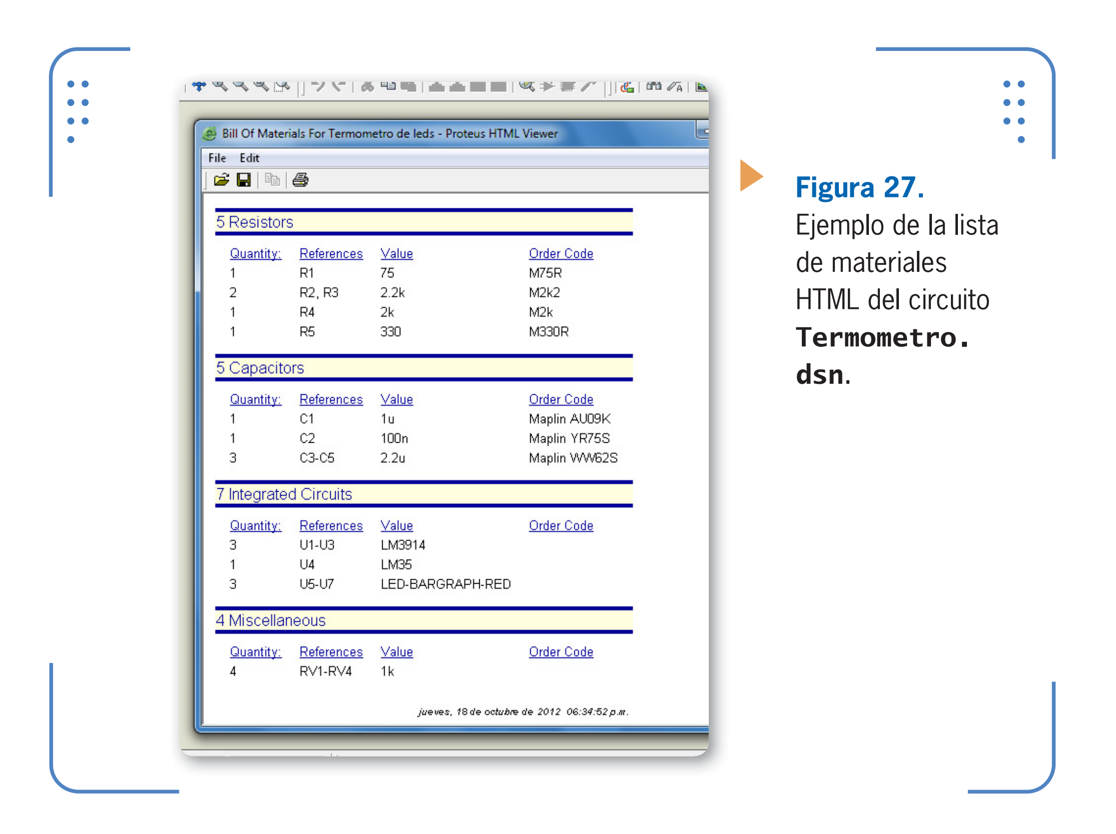


Figura 26. Disponemos de diferentes opciones para generar listas de materiales y componentes.

La opción **HTML** generará un archivo que se puede ver en un navegador web, tal como **Internet Explorer** u **Opera**. El formato **ASCII** es para texto simple, y **CVS** es un formato en el que los valores están separados por comas y que puede ser visto en programas como **Excel** (las comas indican la separación de las columnas). Al seleccionar cualquiera de los formatos para el reporte, se abrirá una ventana, que es un visor HTML que se instala con Proteus, donde observaremos el reporte en el formato elegido.



5 Resistors

Quantity:	References	Value	Order Code
1	R1	75	M75R
2	R2, R3	2.2k	M2k2
1	R4	2k	M2k
1	R5	330	M330R

5 Capacitors

Quantity:	References	Value	Order Code
1	C1	1u	Maplin AU09K
1	C2	100n	Maplin YR75S
3	C3-C5	2.2u	Maplin WW62S

7 Integrated Circuits

Quantity:	References	Value	Order Code
3	U1-U3	LM3914	
1	U4	LM35	
3	U5-U7	LED-BARGRAPH-RED	

4 Miscellaneous

Quantity:	References	Value	Order Code
4	RV1-RV4	1k	

jueves, 18 de octubre de 2012 06:34:52 p.m.

Figura 27. Ejemplo de la lista de materiales HTML del circuito **Termometro.dsn**.

Desde el visor, podemos imprimir la lista de materiales o guardarla. Para esto, hacemos clic en el botón **Save** (que tiene la forma de un pequeño disquete) o en el menú **File/Save as...**. Luego, seleccionamos la carpeta donde queremos almacenar la lista, y quedará guardada en el formato elegido: **.HTML**, **.TXT** o **.CVS**.

Verificador de conexiones eléctricas

Un reporte que puede ser útil para determinar si hay errores en nuestro circuito es el **verificador de conexiones eléctricas**, que genera un informe automático de las conexiones entre los diferentes componentes del circuito y advierte sobre posibles conexiones erróneas o faltantes. Para acceder a este reporte, debemos ir al menú **Tools** y seleccionar **Electrical Rule Check...**.

Se abrirá una ventana de texto que contiene el reporte eléctrico del circuito, en donde veremos algunos datos que hemos colocado en las propiedades del diseño, por ejemplo, el título. Como hemos visto al

aprender a crear o editar componentes, cada pin tiene una asignación de qué función cumple, por ejemplo: salida, entrada, pasivo, etcétera. El verificador revisa la conexión entre diferentes tipos de pines y determina si puede haber un error.

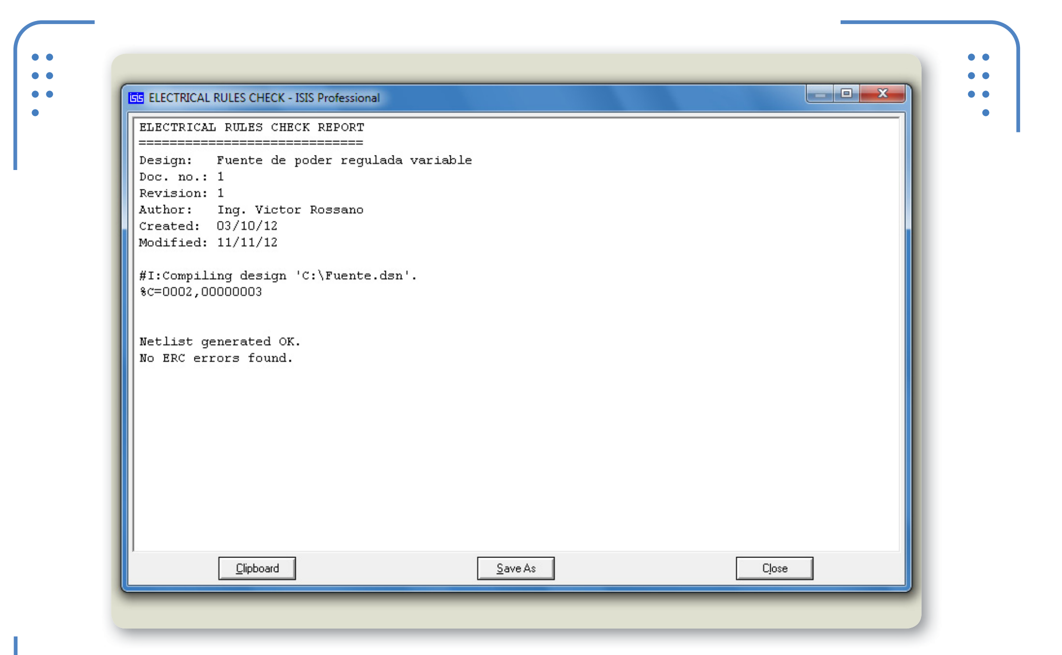


Figura 28. Tenemos un ejemplo del reporte eléctrico del circuito **Fuente.dsn**, que carece de errores.

El reporte del verificador de conexiones eléctricas suele dar dos mensajes principales. Uno de ellos es **ERROR** si encuentra que dos pines clasificados como salidas están conectados, lo cual supondría un cortocircuito o un gran consumo de corriente, o si una terminal Power



CVS COMPACTO VERSUS COMPLETO

Al guardar un archivo **.CVS**, tendremos dos opciones. En **CVS compacto**, los componentes idénticos se agrupan; por ejemplo, si tenemos dos resistores de 10 kOhms, el informe así lo indicará. En **CVS completo** los componentes se listarán uno a uno, sin agrupar los que son idénticos.

está conectada a un pin de salida, casos en los que la conexión podría generar mal funcionamiento o problemas eléctricos graves.

```
ELECTRICAL RULES CHECK REPORT
=============================

Design:   Control de velocidad de un motor por PWM
Doc. no.: 1
Revision: 1
Author:   Ing. Victor Rossano
Created:  11/11/12
Modified: 11/11/12

#I:Compiling design 'C:\PWM.DSN'.
#C=0002,00000003

UNDRIVEN: U1,CV (Input)
UNDRIVEN: $IOSCILLOSCOPE#0011,C (Input)
UNDRIVEN: $IOSCILLOSCOPE#0011,D (Input)

Netlist generated OK.
ERC errors found.
```

Figura 29. Tenemos un ejemplo del reporte eléctrico del circuito **PWM.dsn** con la advertencia **UNDRIVEN**.

El segundo mensaje es solo una advertencia llamada **UNDRIVEN**, que aparecerá cuando haya pines no conectados (flotantes) o clasificados como entradas, unidos, sin ningún pin de salida conectado en esa red. Esto supondría la falta de conexiones al haber pines flotando o una conexión inútil en la que solo hay pines de entrada y nada los alimenta. Veamos algunos ejemplos.

LA ADVERTENCIA UNDRIVEN APARECERÁ CUANDO HAYA PINES NO CONECTADOS



En el reporte del circuito **PWM.dsn** encontramos la advertencia **UNDRIVEN U1, CV (Input)**, que nos dice que la terminal CV del CI 1 no está conectada a ningún lugar. Esto es normal: sabemos que el pin CV del 555 no se conecta en este caso. Esto no es un error real, ni tampoco los otros dos que advierten que los canales C y D del osciloscopio están desconectados.

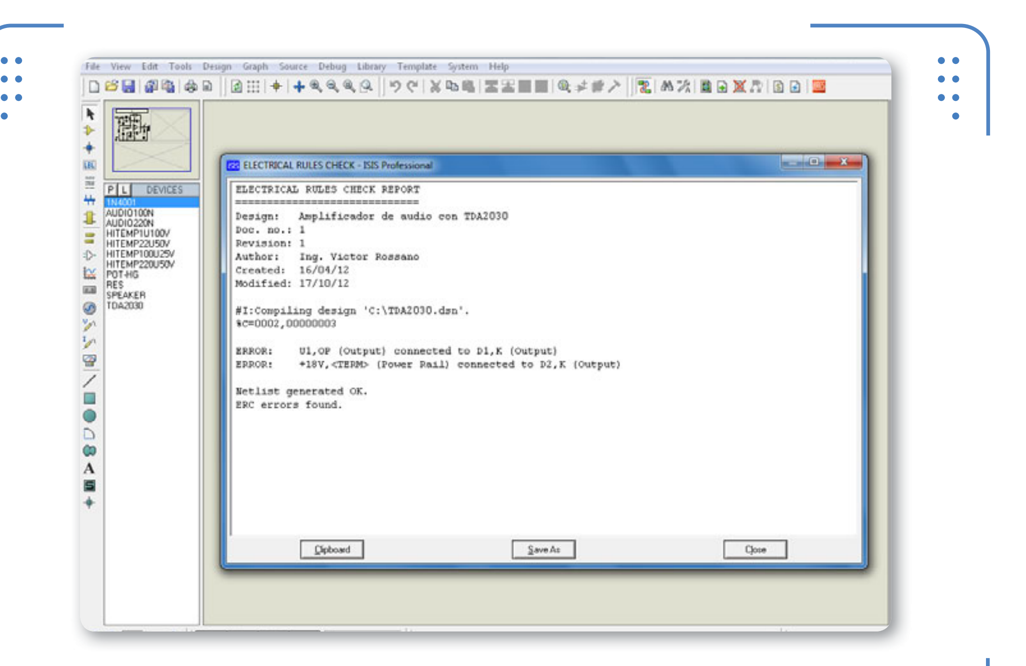


Figura 30. Ejemplo del reporte eléctrico del circuito **TDA2030.dsn**, donde se informan dos errores graves.

En el circuito **TDA2030.dsn**, el verificador informa dos errores graves:
ERROR: U1,OP (Output) connected to D1,K (Output)
ERROR: +18V,<TERM> (Power Rail) connected to D2,K (Output)

Aunque aquí tampoco son reales. El primero indica que la salida del amplificador está conectada a la salida del diodo D1, la terminal del cátodo de los diodos está clasificada como de salida, aunque sabemos que no es así, porque los diodos no tienen salidas. Simplemente, se debe clasificar el pin de alguna forma, y es por eso que el verificador





¿ERRORES O NO?

El reporte del verificador de conexiones eléctricas es solo informativo; no habrá problema alguno al simular un circuito, si es que en él hay errores o advertencias. Tampoco indica si los errores son reales o no, son solo probables. El análisis del reporte queda totalmente bajo criterio del diseñador.

reporta esto como un error. El segundo error es parecido, informa que la terminal Power +18V está conectada a una salida, pero nuevamente se trata del cátodo de un diodo. En estos ejemplos, vemos que no todos los errores reportados deben tomarse en consideración. El verificador de conexiones eléctricas es útil en los casos en que una simulación no funciona de manera adecuada. En él podemos observar si hay pines no conectados por error o conexiones equivocadas.

NO TODOS LOS ERRORES REPORTADOS DEBEN TOMARSE EN CONSIDERACIÓN



El explorador de diseño

Otro reporte que puede ser útil en algunos casos es el explorador de diseño. Para acceder a él, debemos ir al menú **Design** y seleccionar **Design Explorer**; se abrirá una ventana con el explorador. También se puede acceder al explorador de diseño haciendo un clic sobre la barra de estado.

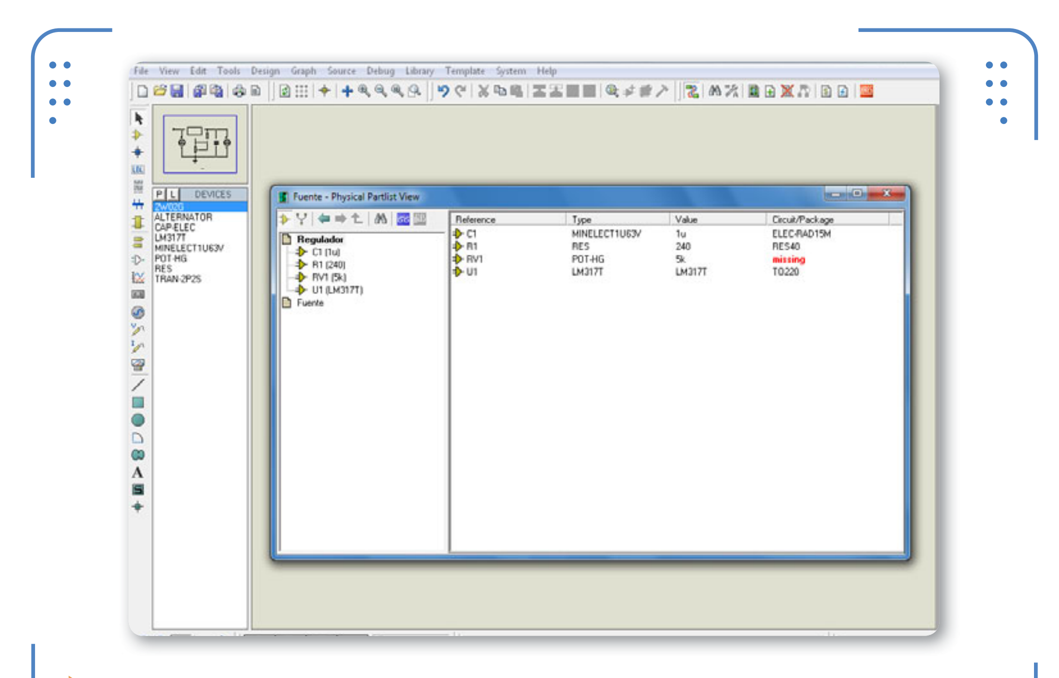


Figura 31. La ventana del explorador de diseño del circuito **Fuente.dsn** con sus hojas y componentes.

La ventana del explorador muestra en el título el nombre del archivo y la leyenda **Physical Partlist View**. Es posible navegar en dos modos. El primero es **componentes**, seleccionando el botón que tiene la forma de un pequeño amplificador operacional amarillo. En este modo el explorador muestra la lista de las hojas y subhojas del diseño en la parte izquierda, y en la derecha, la lista de todos los componentes que contiene la hoja seleccionada y los detalles de cada uno. Podemos hacer un doble clic en una hoja o subcircuito para mostrar la lista de componentes debajo de ella, y en la parte derecha se verá la lista de sus pines y la información de cada uno.

EL EXPLORADOR DE DISEÑOS PERMITE NAVEGAR EN MODO COMPONENTES O REDES



El segundo es **redes**, cuando presionamos el botón con la forma de tres puntos amarillos interconectados. En este modo, en lugar de componentes, la parte derecha mostrará la lista de las redes o nodos de la hoja. Este explorador de diseño es útil para navegar rápidamente por todos los componentes, pines y redes de un circuito, para verificar información tal como los nombres de pines y su tipo, a qué red está conectado un determinado pin, los empaques de los componentes, etcétera.

Se puede acceder de manera sencilla a cualquier componente o red desde el explorador de diseño. Simplemente, hay que seleccionar un componente y presionar el botón **ISIS** de la misma ventana del explorador. Automáticamente esta se cerrará, y nos llevará al componente en la ventana de edición de ISIS. Si seleccionamos cualquier pin o red de conexiones, al presionar el botón **ISIS** nos llevará al lugar donde está esa red, que se mostrará resaltada en rojo.

EL EXPLORADOR DE DISEÑO Y LOS ERRORES

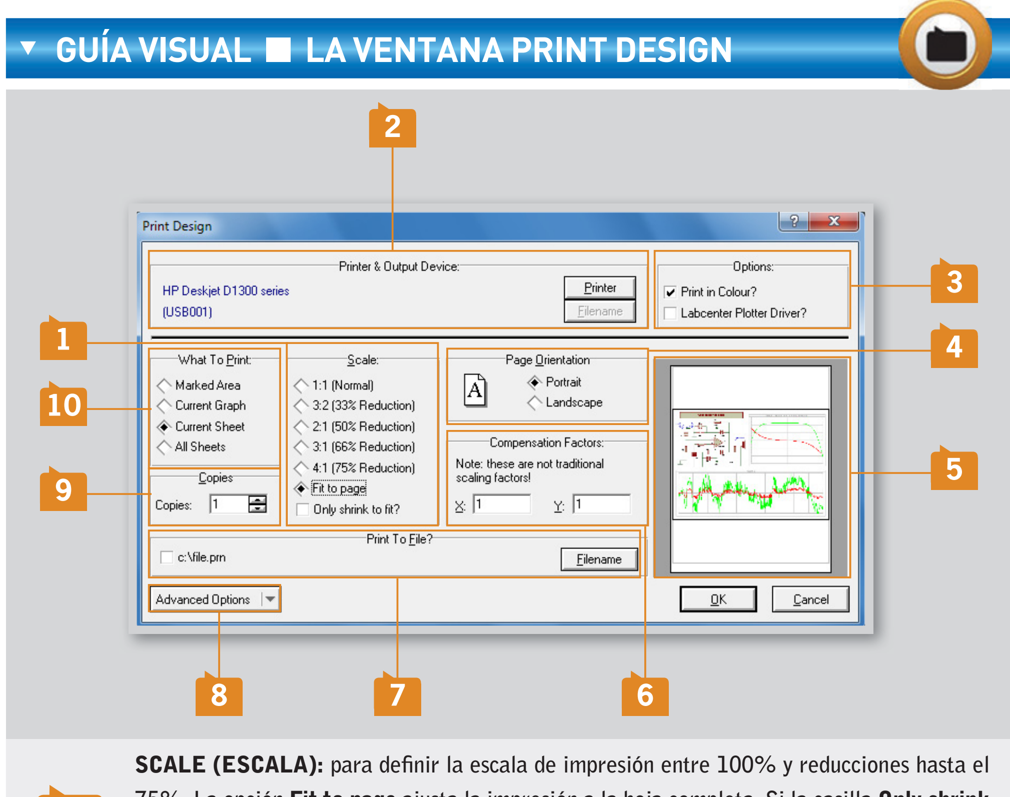
En aquellas situaciones en que tengamos un circuito con errores que no permiten que la simulación comience o continúe, en el informe de simulación se detallarán todos los errores existentes. Si hacemos clic en muchos de ellos, se abrirá el explorador de diseño para indicar en él el componente o la red donde se produjo el error. Esta herramienta de ISIS nos permitirá averiguar rápidamente cuál es el problema que impide la simulación y proceder a su corrección. En el **Capítulo 9**, veremos cómo solucionar los errores.



Imprimir un diagrama

Para imprimir un diagrama, podemos ir al menú **File** y pulsar **Print** o seleccionar el botón con la imagen de una impresora en la barra de opciones de archivo. En ambos casos, se abrirá la ventana **Print Design**. También puede aparecer una pequeña ventana con el mensaje: **Point at the preview and Click your right mouse button for more previewing options**. Esto advierte la posibilidad de hacer clic en la ventana de vista previa con el botón derecho para acceder a más opciones, por ahora vamos a cerrarla. Si marcamos la opción **Don't display this message again**, esta ventana no volverá a aparecer. En la **Guía visual** conoceremos los elementos que podemos configurar.

GUÍA VISUAL ■ LA VENTANA PRINT DESIGN



01 **SCALE (ESCALA):** para definir la escala de impresión entre 100% y reducciones hasta el 75%. La opción **Fit to page** ajusta la impresión a la hoja completa. Si la casilla **Only shrink to fit?** está marcada, solo se ajustará la impresión cuando el dibujo sea muy grande para la hoja. Si la desmarcamos, el dibujo también se ajustará si es más pequeño que la hoja.

02 **PRINTER & OUTPUT DEVICE (IMPRESORA):** define la impresora o el dispositivo que se usará con el botón **Printer**.

03 **OPTIONS (OPCIONES):** algunas opciones extra son:
- **Print in color?:** para imprimir en color.
- **Labcenter Plotter Driver?:** activa el controlador propio para la impresión con plotter.

04 **PAGE ORIENTATION (ORIENTACIÓN DE HOJA):** define la orientación de la hoja, entre vertical (**Portrait**) u horizontal (**Landscape**).

05 **PREVIEW (VISTA PREVIA):** presenta una vista previa de la impresión.

06 **COMPENSATION FACTORS (FACTORES DE COMPENSACIÓN):** para cambiar la escala del dibujo en la impresión. Por ejemplo, si se coloca el valor 2 en ambos, el tamaño del dibujo será el doble. Si se cambia solo un valor, el dibujo estará deformado.

07 **PRINT TO FILE? (IMPRIMIR A ARCHIVO?):** para generar un archivo **.PRN**.

08 **ADVANCED OPTIONS (OPCIONES AVANZADAS):**
- **Alaways Enable Colour Options:** siempre habilitar las opciones de color.
- **Always Enable Labcenter Plotter Support:** siempre habilitar el soporte de Labcenter para plotter.

09 **COPIES (COPIAS):** para elegir el número de copias por imprimir.

10 **WHAT TO PRINT (QUÉ IMPRIMIR):** aquí elegimos qué vamos a imprimir. Las opciones son:
- **Marked Area (área marcada):** se imprimirá solo el área delimitada por la opción **Set Area**.
- **Current Graph (gráfico actual):** se imprimirá solo el gráfico activo, es decir, el que esté marcado en el menú **Graph**, si hay más de uno.
- **Current Sheet (hoja actual):** se imprimirá solo la hoja activa.
- **All Sheets (todas las hojas):** se imprimirán todas las hojas del diseño, si es que hay más de una.

Podemos arrastrar el dibujo en la vista previa para cambiarlo de posición. Al hacer clic en la vista previa con el botón derecho del mouse, aparecerán algunas opciones para posicionar la impresión en la hoja de manera fácil; con ellas podemos ubicarla en la parte superior, inferior, a la izquierda o en el centro. Por último, la opción **Position Output Numerically** abrirá una nueva ventana donde es posible posicionar la impresión con más detalle si lo necesitamos.



Exportar circuitos en formatos gráficos

También es posible exportar los circuitos o parte de ellos en formatos gráficos diferentes, para lo cual en el menú **File** hay una opción llamada **Export Graphics**.

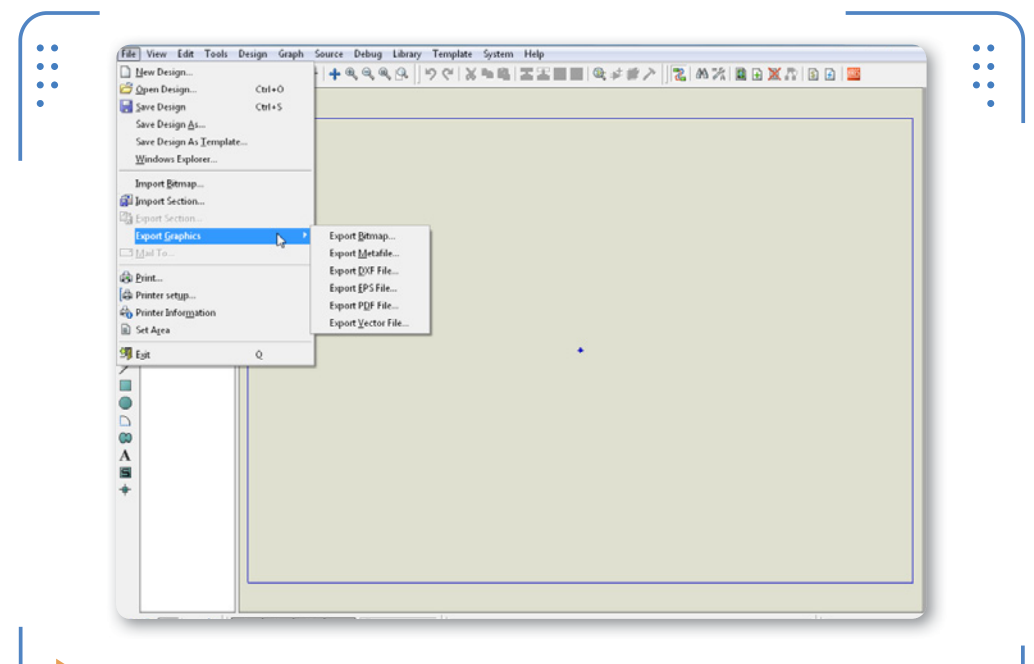


Figura 32. Los circuitos que diseñemos se pueden exportar como imágenes gráficas de diferente tipo.





LA OPCIÓN SET AREA

En el menú **File** hay una opción llamada **Set Area**, a la cual se puede acceder también desde el botón **Mark Output Area** en la barra de opciones de archivo. Esta permite marcar el área deseada con un cuadro de color gris. Luego, si elegimos la opción **Marked Area** en la ventana de impresión, solo se imprimirá el área que se encuentra delimitada por este cuadro.

Como podemos apreciar en la **Figura 32**, hay seis opciones diferentes de exportación. La más común es exportar una imagen de mapa de bits o **.BMP**. Al elegir **Export Bitmap**, aparecerá una ventana con el mismo nombre en donde podemos configurar los parámetros para exportar la imagen. En la siguiente **Guía visual** los conoceremos en detalle.

GUÍA VISUAL ■ VENTANA EXPORT BITMAP

01 **SCOPE (ÁREA):** aquí definimos el área que se exportará como imagen; son las mismas opciones que las de impresión.

02 **RESOLUTION (RESOLUCIÓN):** para indicar la resolución de la imagen generada, desde 100 hasta 600 puntos por pulgada. A mayor resolución, más detallada será la imagen, pero más pesado será el archivo generado.

03 **COLOURS (COLORES):** permite establecer cuántos colores se utilizarán:
- **Mono:** un solo color (blanco y negro).
- **16:** una paleta de 16 colores.
- **256:** una paleta de 256 colores.
- **Display:** una paleta de colores basada en el controlador gráfico del sistema, se obtendrán los colores originales de cada elemento.



ROTATION (ROTACIÓN): para rotar la imagen en la orientación deseada, vertical u horizontal.



BACKGROUND (FONDO): permite cambiar el color de fondo, por defecto se tendrá el color de fondo original. Esta opción está desactivada en el modo Mono, ya que en este el fondo siempre será de color blanco.



OUTPUT TO FILE? (SALIDA A ARCHIVO): para especificar si la imagen se guardará en un archivo. Si presionamos el botón **Filename**, elegimos el nombre del archivo y el lugar para guardarlo. Si la casilla está desmarcada, la imagen generada solo se copiará en el Portapapeles de Windows.

Además de mapa de bits, tenemos opciones para exportar los circuitos en otros formatos: un meta-archivo de Windows **.WMF**, un archivo **.DXF** que puede ser abierto en AutoCAD, un archivo **.EPS** que puede ser abierto por programas como Illustrator, un archivo **.PDF** o un archivo vectorial **.HGL**. Las opciones para estos formatos son muy parecidas a las del mapa de bits.



RESUMEN

Proteus cuenta con gran cantidad de herramientas para mantener ordenados los diagramas, generar reportes, y analizar los diseños y sus componentes. En este capítulo, aprendimos a dibujar diseños en múltiples hojas o utilizando subcircuitos. Vimos cómo modificar los componentes disponibles o crear nuevos símbolos desde cero. También conocimos el procedimiento para exportar listas de componentes o imágenes de los circuitos y cómo imprimir un diseño. Todas estas herramientas expanden las posibilidades de manejar los diagramas dibujados en ISIS.

Actividades

TEST DE AUTOEVALUACIÓN

1	¿Cómo se agrega una nueva hoja a un diseño?
2	¿Cómo se cambia de hoja en un circuito que posee más de una?
3	¿Cómo se dibuja un subcircuito?
4	Un componente está formado por elementos gráficos y _____.
5	Para crear un nuevo componente se invoca el comando __________.
6	Para modificar un componente se utiliza el comando _______.
7	¿Dónde podemos agregar el nombre del autor del diseño?
8	¿Cómo se genera una lista de materiales (BOM)?
9	¿Para qué sirve el verificador de conexiones eléctricas?
10	¿Cómo se exporta una imagen BMP de un diseño?

ACTIVIDADES PRÁCTICAS

1	Busque un circuito que pueda ser dibujado en múltiples hojas y dibújelo en ISIS.
2	Busque y dibuje un circuito usando uno o más subcircuitos.
3	El circuito integrado LM75 no se encuentra en Proteus; busque información sobre él y créelo como un nuevo componente.
4	Descargue el archivo **Teclado.dsn**, y siguiendo lo descripto en la sección Un teclado personalizado, modifique el teclado.
5	Abra en ISIS cualquier archivo de un diseño que planee construir físicamente, genere una lista de materiales e imprímala para poder comprar los componentes requeridos.





Personalización de la interfaz de ISIS

Proteus admite un buen nivel de personalización de la interfaz de ISIS. Permite cambiar los colores de los diseños y también el estilo de los textos. Es posible crear plantillas para agregar estilos personalizados a los nuevos diseños, y algunos elementos, como las barras de herramientas, pueden reposicionarse dentro de la ventana de ISIS. En este capítulo nos referiremos a todas estas opciones de personalización.



↗↗↗

Las barras de herramientas

Las barras de herramientas tienen una línea, ya sea en el lado izquierdo para las horizontales o en la parte superior para las verticales. Si colocamos el cursor del mouse en ella, este tomará la forma de una flecha doble que indica que podemos mover esa barra de herramientas a otro lugar, si lo deseamos.

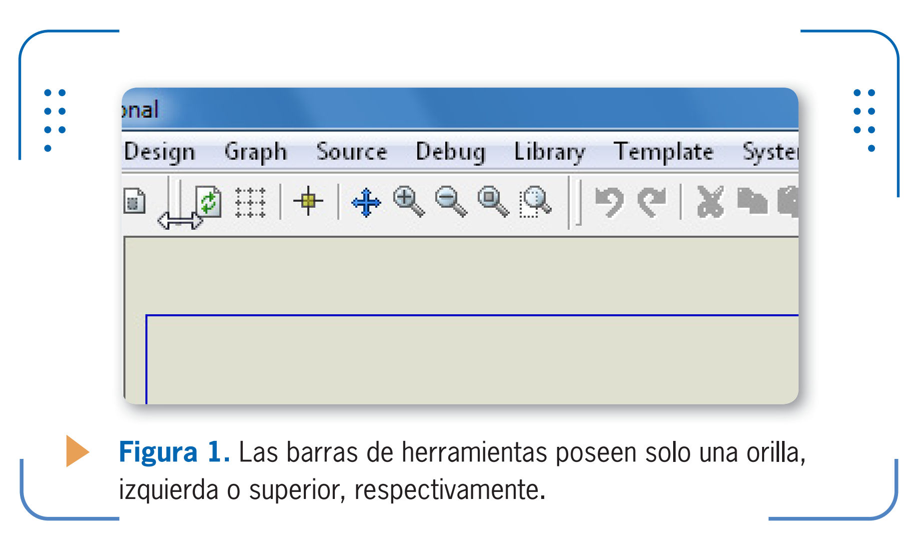
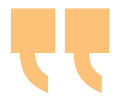


Figura 1. Las barras de herramientas poseen solo una orilla, izquierda o superior, respectivamente.

Al hacer clic en dicha línea, es posible arrastrar la barra para cambiarla de posición o desplazarla dentro de la misma área donde se encuentra para variar su ubicación con respecto a las demás barras. Por ejemplo, podemos trasladar la barra de opciones de archivo hacia la derecha para que sea la última barra de la parte superior de la pantalla, si es que nos resulta mejor tenerla en ese lugar. También es posible arrastrarla hacia cualquiera de los cuatro lados de la ventana para moverla ahí, ya sea a la izquierda, a la derecha, abajo o arriba. Al arrastrar una de las barras a otra orilla de la ventana, el cursor tomará la forma de una pequeña mano, indicando que podemos ubicarla en ese lugar al soltar el botón del mouse. Si colocamos una barra en un lugar en el cual ya existen otras barras y el espacio no

PODEMOS MODIFICAR LA UBICACIÓN DE LAS BARRAS DE HERRAMIENTAS



es suficiente para todas, la nueva barra se colocará en una segunda línea, ya sea debajo o a la derecha de la otra. El cambio de posición de las barras de herramientas depende por completo de una decisión personal. Esta posibilidad nos permitirá adaptar nuestro entorno de trabajo para trabajar con la mayor comodidad.

Ocultar las barras de herramientas

Si deseamos ocultar una o más barras de herramientas, debemos ir al menú **View** y pulsar **Toolbars...**. Con esto se abrirá una pequeña ventana con el título **Show/Hide Toolbars**.

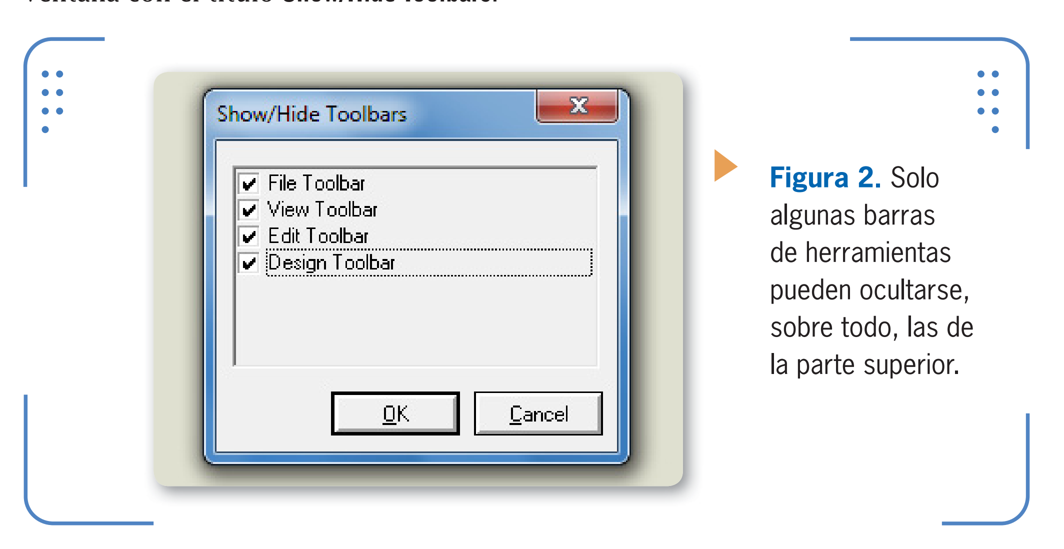


Figura 2. Solo algunas barras de herramientas pueden ocultarse, sobre todo, las de la parte superior.

En esta ventana tenemos una lista de algunas de las barras de herramientas; podemos ocultar o mostrar cualquiera de ellas simplemente marcando o desmarcando la casilla correspondiente.



ALGUNAS BARRAS NO SE OCULTAN

En la ventana **Show/Hide Toolbars** solo podemos ver algunas de las barras de herramientas, las demás, las que no se incluyen en ella, no pueden ocultarse, por ejemplo, la barra de herramientas de **Modo** o la de **Simulación**. Esto se debe a que los botones o funciones de estas barras no aparecen en los menús, y si las ocultamos, no sería posible usar esos comandos.



El Selector de objetos

También es posible cambiar la posición o el tamaño del **Selector de objetos**. Al colocar el cursor sobre la orilla del selector, este tomará la forma de una flecha doble; si hacemos clic y arrastramos, podemos redimensionarlo. Si continuamos arrastrando hasta el costado opuesto, el **Selector de objetos** se posicionará en ese lado de la ventana.

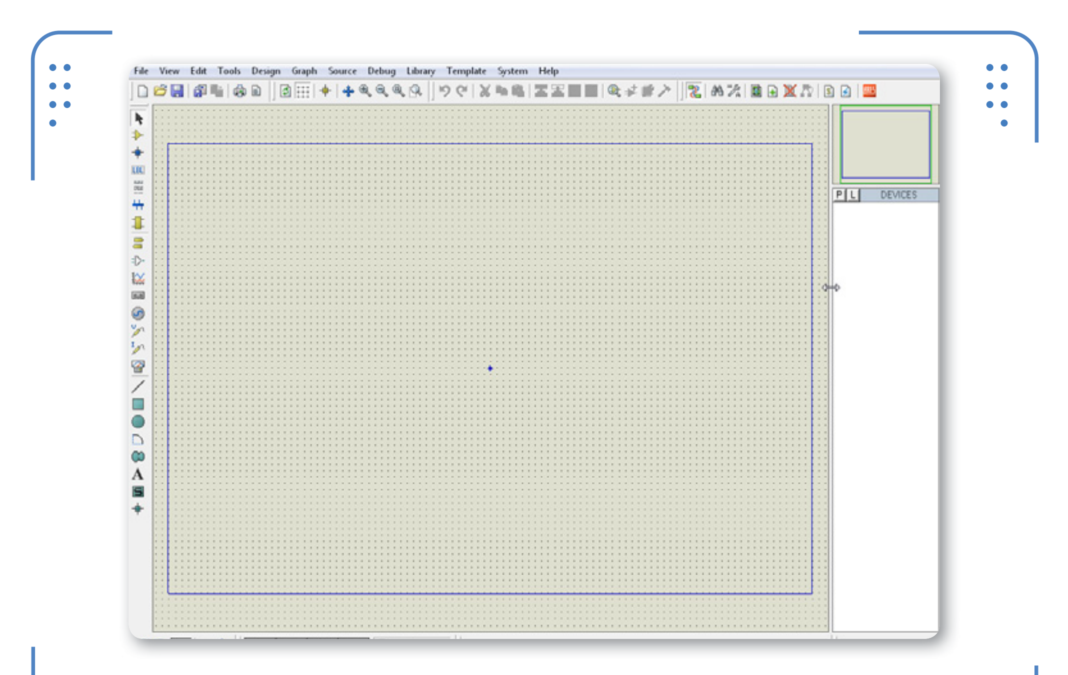




Figura 3. El **Selector de objetos** puede colocarse en el lado opuesto si lo deseamos, con solo arrastrarlo hacia ese lugar.



REDIMENSIONAR LA VISTA PREVIA

La **ventana de vista previa** del **Selector de objetos** es otro de los elementos que también puede redimensionarse. Si colocamos el cursor del mouse en su barra inferior, y arrastramos hacia abajo, lograremos hacerla más alta. Esto puede ser útil si ya redimensionamos el **Selector de objetos** y queremos hacerla cuadrada en lugar de rectangular, para posicionarla adecuadamente en la ventana.

Al soltar el botón del mouse, quedará en su nueva posición o tamaño. El **Selector de objetos** solo puede colocarse en el lado derecho o izquierdo de la ventana de ISIS.

Ocultar el Selector de objetos

Podemos hacer que el **Selector de objetos** se oculte automáticamente cuando no estamos utilizándolo. Para esto, debemos hacer un clic con el botón derecho sobre el selector, en cualquier modo, y en el menú contextual seleccionar la opción llamada **Auto Hide**.

Figura 4. La opción **Auto Hide** oculta el **Selector de objetos** hasta que lo utilicemos otra vez.

De esta manera, al estar trabajando en la ventana de edición, el **Selector de objetos** se ocultará de forma automática y en su lugar se mostrará solo una barra vertical, con el nombre del modo en que nos encontremos en ese momento y el objeto seleccionado.

Al colocar el cursor sobre dicha barra o presionar cualquier botón en la barra de herramientas de **Modo**, el **Selector de objetos** volverá a

aparecer para poder usarlo. Si queremos que el **Selector de objetos** se muestre siempre de manera estática, debemos desmarcar la opción **Auto Hide** en el menú contextual.

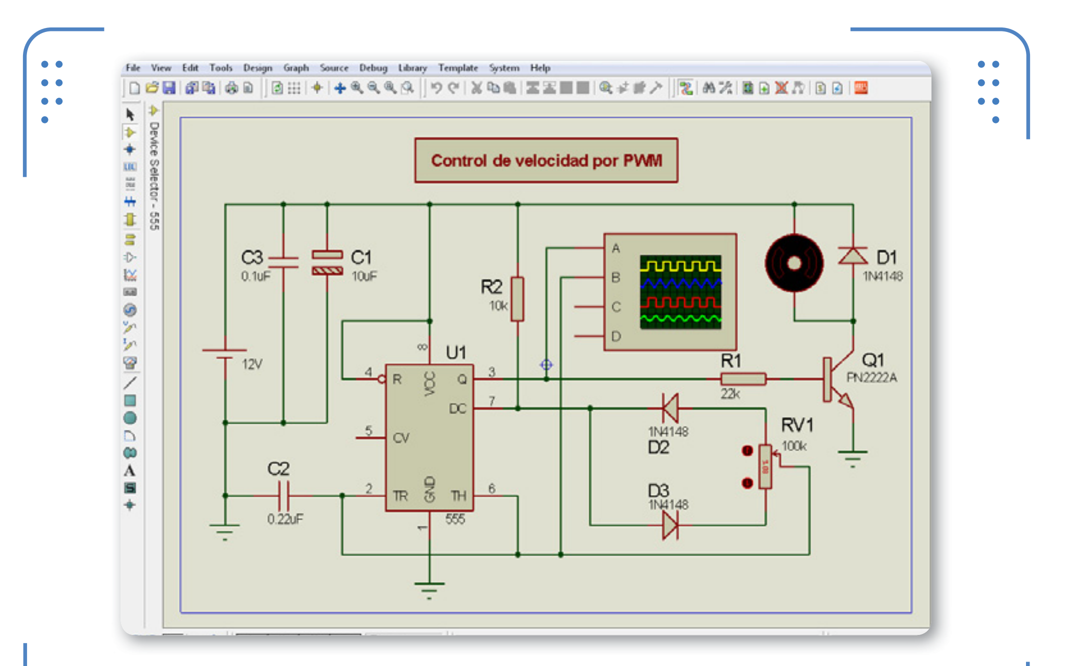


Figura 5. El **Selector de objetos** se oculta para tener un área de trabajo mayor en la ventana de edición.



Cuadro de datos (header)

En el **Capítulo 1** vimos la posibilidad de tener un cuadro de datos en nuestros diseños, el cual se genera cuando creamos un nuevo archivo mediante el menú **File** y seleccionamos la opción **New Design...**. Los datos que contiene son:

- **FILE NAME**: es el nombre del archivo **.DSN**.
- **DESIGN TITLE**: es el título del diseño.
- **PATH**: presenta la ruta completa de la ubicación del archivo **.DSN**.
- **BY**: es el nombre del autor.
- **REV**: número de revisión del circuito.

- **DATE**: muestra la fecha de modificación.
- **PAGE**: indica el número de la hoja activa y el total de hojas del diseño.
- **TIME**: informa la hora de modificación del archivo.



Figura 6. El cuadro de datos predeterminado también se encuentra en la librería de símbolos con el nombre HEADER.

La información de fecha y hora se toma del reloj del sistema; así como el nombre del archivo, la ruta donde se ubica, el autor, la revisión y el título se toman de las propiedades del diseño (menú **Design/Edit Design Properties...**) al completar los campos correspondientes. Si el cuadro de datos por defecto no es de nuestro agrado, podemos modificarlo o crear uno. A continuación veremos cómo hacerlo.

SI EL CUADRO DE DATOS POR DEFECTO NO ES DE NUESTRO AGRADO, PODEMOS CREAR UNO NUEVO

La hoja global

Los cuadros de datos de las plantillas existentes se encuentran en la **hoja global**, es por eso que desde las hojas del diseño no podemos seleccionarlos ni modificarlos. Si vamos al menú **Template** y pulsamos **Goto Master Sheet**, accederemos a la hoja global, la cual se usa como fondo en todas las hojas del diseño, es decir, lo que está en ella aparecerá en todas las demás. Al estar en la hoja global, podemos seleccionar y modificar el cuadro de datos y el marco de la plantilla o crear una desde cero si lo deseamos.

El formato del cuadro de datos

El cuadro de datos y el marco están formados por elementos gráficos (cuadros, líneas) y por texto simple (texto en 2D). Además, es posible usar algunas expresiones especiales para invocar las propiedades que se completan automáticamente, por ejemplo, **@FILENAME** para invocar en ese lugar el nombre del archivo. En la **Tabla 1** detallamos la lista completa de las expresiones.

EXPRESIONES

PUERTO	DESCRIPCIÓN
@DTITLE	Título del diseño, tomado de las propiedades del diseño.
@STITLE	Título de la hoja, tomado de las propiedades de la hoja.
@DOCNO	Número de documento, tomado de las propiedades del diseño.
@REV	Número de revisión, tomado de las propiedades del diseño.
@AUTHOR	Nombre del autor, tomado de las propiedades del diseño.
@CDATE	Fecha de creación del archivo.
@MDATE	Fecha de modificación del archivo.
@WS_CDATE	Fecha de creación del archivo, en formato corto de Windows.
@WL_CDATE	Fecha de creación del archivo, en formato largo de Windows.
@WS_MDATE	Fecha de modificación del archivo, en formato corto de Windows.
@WL_MDATE	Fecha de modificación del archivo, en formato largo de Windows.
@CTIME	Hora de creación del archivo en el formato de Windows.
@MTIME	Hora de modificación del archivo en el formato de Windows.
@PAGENUM	Número de hoja.
@PAGECOUNT	Número de hojas del diseño.

@PAGE	Hoja en formato X/Y, donde X es el número de hoja, e Y, el número total de hojas.
@FILENAME	Nombre del archivo.
@PATHNAME	Nombre del archivo con la ruta completa de su ubicación en el sistema.

Tabla 1. Expresiones para extraer datos de forma automática.

Las expresiones deben ubicarse solas en un **texto 2D**. Por ejemplo, si colocamos un nuevo texto 2D y escribimos **hoja @PAGENUM de @ PAGECOUNT**, esto no funcionará. Tenemos que escribir cada elemento en un texto 2D por separado y luego situar uno junto al otro. Así podemos modificar un cuadro de datos existente, o crear uno con el tamaño, la apariencia, el color y los textos deseados.

Ejemplo

Para ver un ejemplo de un cuadro de datos personalizado, debemos descargar el archivo **FuenteHeader.dsn.** Hemos agregado uno al archivo de la **fuente variable** que vimos en el capítulo anterior.

En este ejemplo hemos colocado el cuadro de datos en la hoja global, aunque también podemos ubicarlo en una de las hojas, si deseamos que aparezca solo en una hoja y no en todas. Debemos tomar en cuenta que si redimensionamos la hoja de trabajo dentro de la hoja global, esto se aplicará a todas las hojas y subhojas del diseño, modificación que puede ser útil en algunos casos pero no en otros.

Mientras estemos trabajando en la hoja global, no veremos los circuitos ni los componentes que se encuentran en la ventana de edición, ya que la hoja global no permite colocar algunos elementos, como componentes, terminales, sondas, etcétera. Además, notaremos que al estar en la hoja global, muchos de los botones de la barra de herramientas de **Modo** se encontrarán desactivados. Para salir de la hoja global, simplemente debemos elegir cualquier hoja del diseño, ya sea mediante el menú contextual al hacer un clic derecho en un lugar vacío o a través del menú **Design**.

AL TRABAJAR EN LA HOJA GLOBAL, NO VEREMOS LOS CIRCUITOS NI LOS COMPONENTES



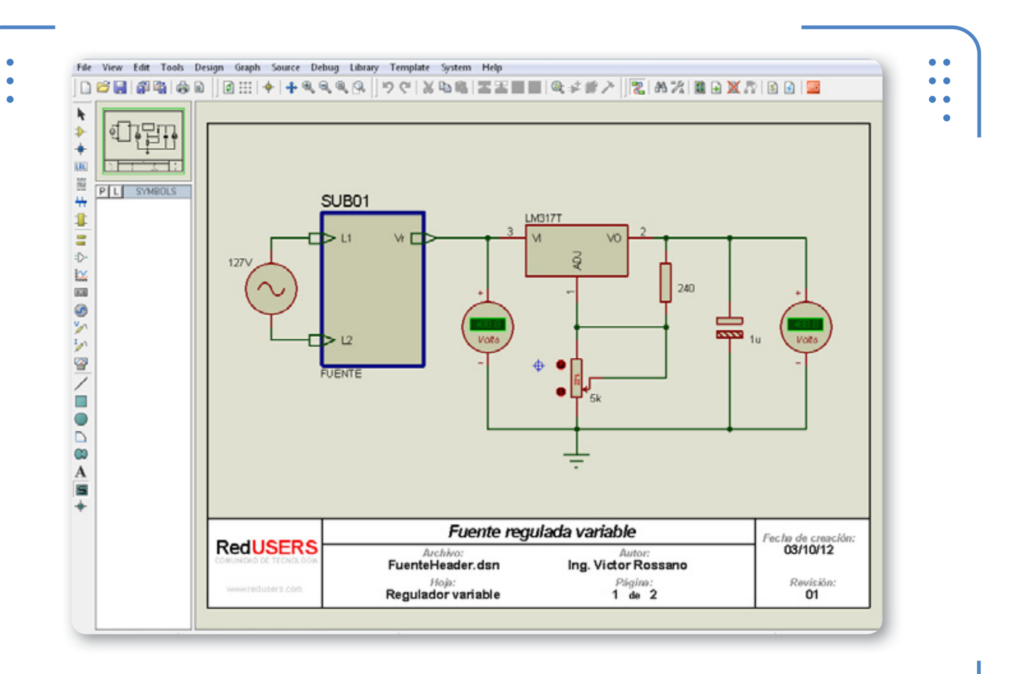


Figura 7. Un cuadro de datos personalizado puede brindar profesionalismo a nuestros diseños de forma simple.



Estilos gráficos y de texto

Además de las posibilidades de modificación de la interfaz que vimos hasta ahora, el módulo ISIS también permite personalizar los colores y textos de los elementos que aparecen en la ventana de edición, para ajustarlos a nuestro gusto si lo deseamos. En las siguientes secciones veremos cómo hacerlo.



NOTAS DE TEXTO EN CUADROS DE DATOS

En los cuadros de datos también podemos usar notas de texto (**Text Script Mode**), aunque en estas las expresiones no funcionan. Para la expresión **@DTITLE**, si el campo respectivo en las propiedades del diseño está vacío, en su lugar aparecerá la ruta completa del archivo, y en los demás no aparecerá nada.

Los estilos globales y locales

Todos los elementos que se colocan en la ventana de edición poseen un estilo definido; por ejemplo, las líneas de conexión tienen por defecto el mismo color, estilo, grosor y un tipo de texto para sus etiquetas, esto se conoce como **estilo global**. En algunos elementos es posible cambiar estas características de forma individual, es decir, solo la apariencia de ese elemento dentro de un diseño determinado. Por ejemplo, si vamos a las propiedades de una línea de conexión, podemos cambiar su color y grosor. Al hacerlo, notaremos que el estilo solo se aplicará a esa línea, porque lo hemos cambiado de forma local.

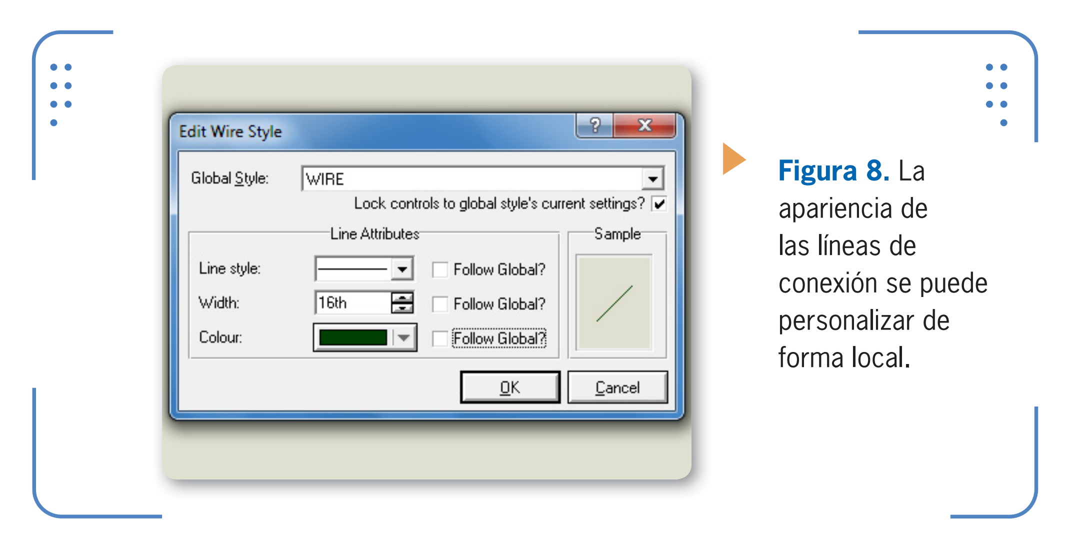


Figura 8. La apariencia de las líneas de conexión se puede personalizar de forma local.

En la **Figura 8** se muestra la ventana de propiedades de una línea de conexión de un diseño cualquiera. Podemos ver que en el campo **Global Style** está seleccionado el estilo **WIRE**, que es la apariencia global de todas las líneas de conexión por defecto. En la sección **Line Attributes** tenemos tres opciones, cada una con un cuadro que dice **Follow Global?**. Si este se encuentra marcado, el estilo de la línea de conexión seguirá el estilo global; si lo desmarcamos, entonces podremos cambiar ese parámetro de forma local. También es posible elegir un estilo global de la lista **Global Style** para aplicarlo a esa línea de conexión.

Algunos elementos solo admiten ser cambiados de forma global, como los bloques de subcircuito, que no pueden ser configurados de manera local. Si queremos cambiar su estilo, debemos hacerlo de forma global, y todos los subcircuitos dentro de un diseño serán siempre iguales.

Los estilos del diseño

En el menú **Template** tenemos opciones para cambiar los estilos gráficos de diferentes elementos. Una de ellas es **Set Design Defaults**, que permite modificar los estilos de la ventana de edición y de las animaciones. En la siguiente **Guía visual** conoceremos sus elementos.

▼ GUÍA VISUAL ■ VENTANA ESTILOS DEL DISEÑO

01 **COLOURS:** aquí se definen los colores de la ventana de edición:
- **Paper Colour:** color del papel o fondo.
- **Grid Colour:** color de la rejilla.
- **Work Area Box Colour:** color del cuadro que muestra la posición en la hoja en la ventana de vista previa.
- **World Box Colour:** color del borde de la hoja.
- **Highlight Colour:** color de los elementos seleccionados.
- **Drag Colour:** color de los elementos mientras son arrastrados de un lugar a otro.

02 **ANIMATION:** aquí se definen los colores de la animación de los circuitos durante las simulaciones:
- **Positive Colour:** color para voltajes positivos.

- **Ground Colour:** color para tierra (GND).
- **Negative Colour:** color para voltajes negativos.
- **Logic '1' Colour:** color para el nivel lógico '1'.
- **Logic '0' Colour:** color para el nivel lógico '0'.
- **Logic '?' Colour:** color para el nivel lógico indefinido.



FONT FACE FOR DEFAULT FONT: aquí podemos elegir la fuente que será utilizada en todos los textos configurados para usar la fuente por defecto.



RENDER MASTER SHEET?: para mostrar u ocultar la hoja global; si desactivamos la hoja global, no se mostrará en ninguna hoja del diseño.

05

HIDDEN OBJECTS (OBJETOS OCULTOS): aquí elegimos si se muestran o no los elementos ocultos:

- **Show hidden text?:** para mostrar u ocultar los textos ocultos de los componentes.
- **Show hidden pins?:** para mostrar u ocultar los pines ocultos de los componentes.
- **'Hidden' Colour:** permite definir un color para los elementos ocultos, si se muestran.

Cuando hacemos un clic en alguno de los campos que muestran el color actual de cada elemento, aparecerá una paleta de colores con diferentes opciones para seleccionar uno nuevo, o un botón en la parte inferior llamado **Other...** con el cual es posible elegir un color personalizado. De esta manera tenemos acceso a una gran variedad de colores para combinarlos a nuestro gusto.

El texto oculto (**Show hidden text?**) se refiere a las propiedades del componente, con excepción del valor y la referencia, que estarán visibles en el diagrama junto a cada componente. Pero si todas están marcadas como ocultas en sus propiedades, en su lugar se mostrará **<TEXT>** para indicar que hay textos ocultos. Desmarcando la opción **Show hidden text?** podemos ocultar todas estas etiquetas en el diagrama, que pueden resultar algo molestas.

EL VALOR Y LA REFERENCIA SE VERÁN JUNTO A CADA COMPONENTE EN EL DIAGRAMA



Si marcamos la casilla para mostrar los pines ocultos, estos estarán visibles en el diagrama en color gris, pero no podremos conectarlos como los demás pines, sino que seguirán usándose de la misma forma de siempre, como lo estudiamos en el **Capítulo 3**.

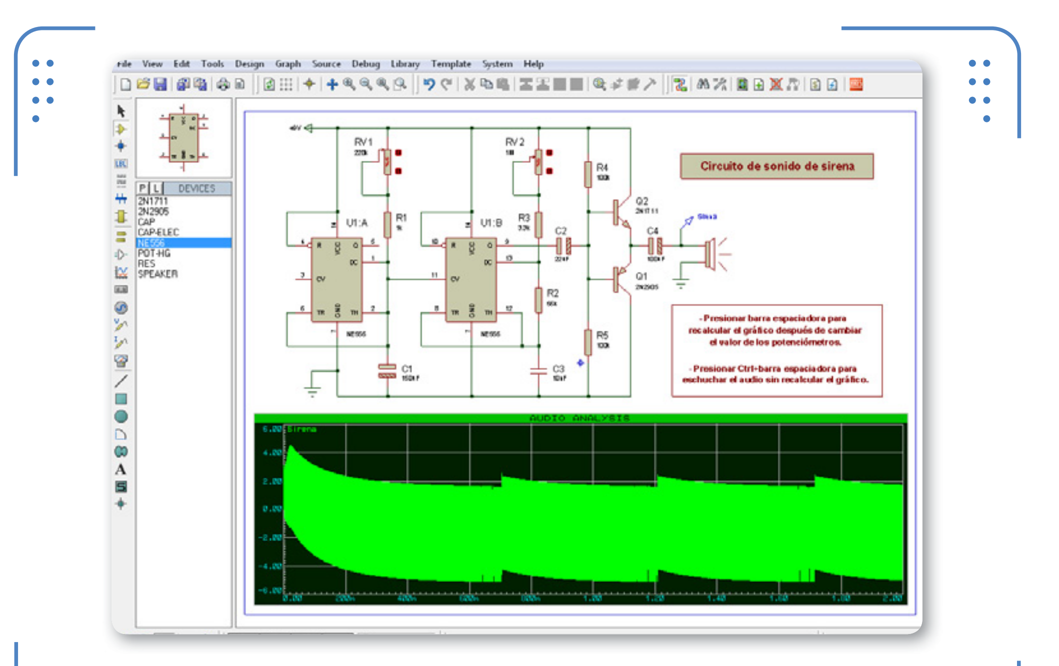


Figura 9. Ejemplo de un diseño con el fondo cambiado a color blanco desde los estilos del diseño.

Colores de los gráficos de simulación

Los gráficos de simulación también pueden personalizarse de forma global para modificar su apariencia.

Para hacerlo, debemos ingresar al menú **Template** y, luego, pulsar la opción **Set Graph Colours...**. A continuación, se abrirá la ventana denominada **Graph Colour Configuration**, que explicaremos en profundidad en la siguiente **Guía visual**.



LOGOS EN CUADROS DE DATOS

Es posible importar una imagen si deseamos colocar un logo en el cuadro de datos. Para esto, tenemos que ingresar al menú **File** y seleccionar la opción **Import Bitmap...**. Pero debemos tomar en cuenta que el único formato permitido es el **.BMP** (bitmap). Estas imágenes son algo pesadas y pueden hacer que el tamaño del archivo **.DSN** también sea grande al guardarlo.

GUÍA VISUAL ■ COLORES DE LOS GRÁFICOS

01 **GENERAL APPEARANCE:** aquí se definen los colores de la ventana de los gráficos de simulación:

- **Graph outline:** borde y rejilla del gráfico.
- **Background:** fondo.
- **Graph Title:** barra de título.
- **Graph Text:** texto que aparece en los ejes del gráfico.
- **Tagged/Hilite:** elementos seleccionados dentro del gráfico, por ejemplo, cuando seleccionamos la etiqueta de una señal en el gráfico.

02 **ANALOGUE TRACES:** aquí especificamos los colores de las señales analógicas en los gráficos.

03 **DIGITAL TRACES:** aquí elegimos los colores de las señales digitales en los gráficos.

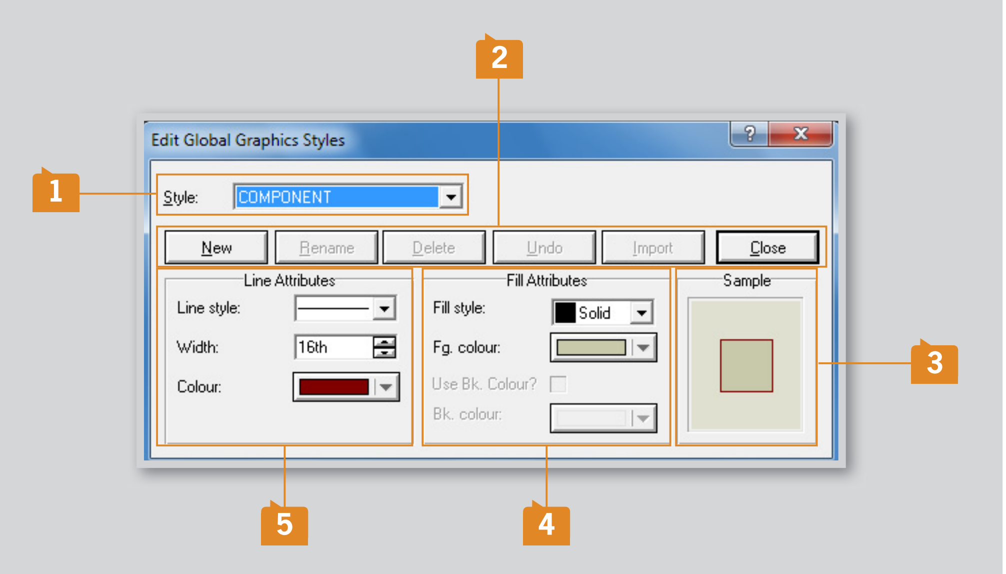
Los estilos gráficos globales

Si ingresamos al menú **Template** y seleccionamos la opción denominada **Set Graphics Style...**, accedemos a las diferentes opciones para cambiar de forma global los colores de distintos elementos.

En la siguiente **Guía visual** explicaremos cada una de ellas en detalle para poder modificar los colores.

▼ GUÍA VISUAL ■ VENTANA DE COLORES GLOBALES





01 **STYLE:** para elegir el elemento que vamos a modificar. En esta lista encontramos todos los elementos que pueden personalizarse: pines, bloques de subcircuito, generadores, sondas, terminales, líneas de conexión, etcétera.

02 **BOTONES:** ayudan con la personalización:

- **New:** crea un nuevo estilo.
- **Rename:** renombra un estilo existente.
- **Delete:** borra un estilo existente.
- **Undo:** revierte los cambios realizados en el estilo seleccionado.
- **Import:** para importar un estilo.
- **Close:** cierra la ventana y aplica todos los cambios hechos en ella.

Algunos estilos no pueden ser borrados ni renombrados, en cuyo caso los botones **Rename** y **Delete** estarán desactivados.

03 **SAMPLE:** presenta una muestra de la apariencia de los colores elegidos.

04 **FILL ATTRIBUTES:** define los atributos de relleno, si es sólido, con patrones o vacío (**Fill Style**), y el color (**Fg color**).

05 **LINE ATTRIBUTES:** para seleccionar el tipo de línea, color y grosor.

Desde esta ventana personalizamos los colores para los diferentes elementos; por ejemplo, si en **Style** elegimos el elemento **COMPONENT**, cambiaremos el color de los cuerpos de todos los componentes. Como son estilos globales, se aplican en todos los componentes ya ubicados en la ventana de diseño y en los que utilicemos en el futuro.

Los estilos de texto globales

Al igual que con colores, también podemos personalizar los textos de forma global, por ejemplo, el estilo en que se muestran las etiquetas de todos los componentes. Para hacerlo, debemos ingresar al menú **Template** y seleccionar la opción **Set Text Styles...**.

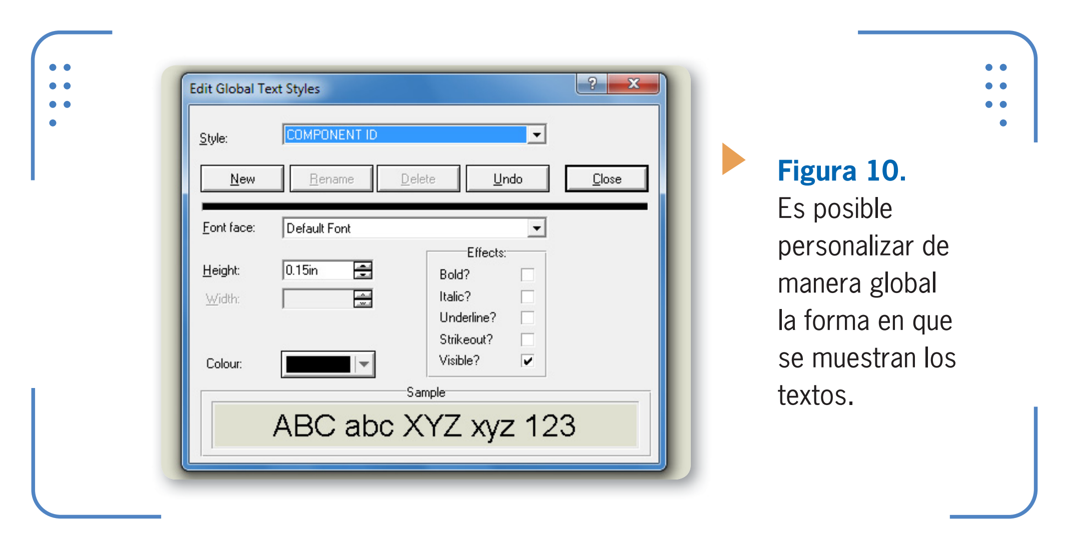
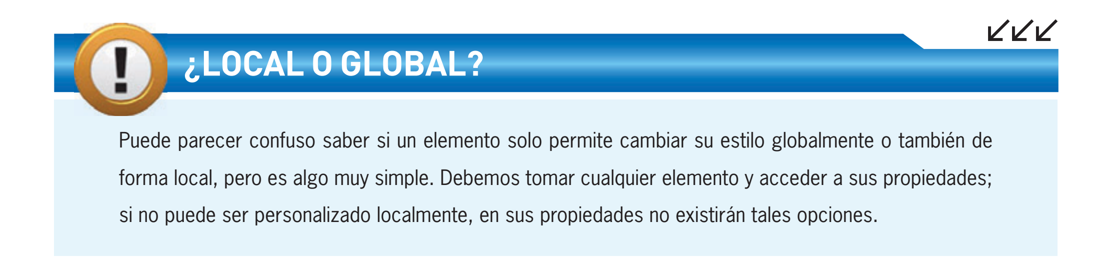


Figura 10. Es posible personalizar de manera global la forma en que se muestran los textos.

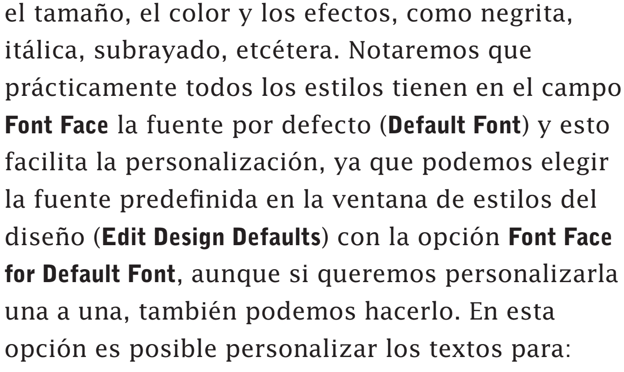
La ventana es muy similar a la de los colores globales, solo que aquí debemos definir los parámetros de texto para cada estilo del campo **Style**. En el campo **Font Face** elegimos la fuente; y en los demás,



¿LOCAL O GLOBAL?

Puede parecer confuso saber si un elemento solo permite cambiar su estilo globalmente o también de forma local, pero es algo muy simple. Debemos tomar cualquier elemento y acceder a sus propiedades; si no puede ser personalizado localmente, en sus propiedades no existirán tales opciones.

TODOS LOS ESTILOS TIENEN EN EL CAMPO FONT FACE LA FUENTE POR DEFECTO



el tamaño, el color y los efectos, como negrita, itálica, subrayado, etcétera. Notaremos que prácticamente todos los estilos tienen en el campo **Font Face** la fuente por defecto (**Default Font**) y esto facilita la personalización, ya que podemos elegir la fuente predefinida en la ventana de estilos del diseño (**Edit Design Defaults**) con la opción **Font Face for Default Font**, aunque si queremos personalizarla una a una, también podemos hacerlo. En esta opción es posible personalizar los textos para: valor e identificador de componentes, nombre y número de pines, etiquetas de terminales, líneas de conexión, sondas, etcétera.

Estilo del texto en 2D

Desde el menú **Template**, al seleccionar la opción **Set Graphics Text**, podemos personalizar el estilo global del texto en 2D, perteneciente a las herramientas de dibujo en 2D.

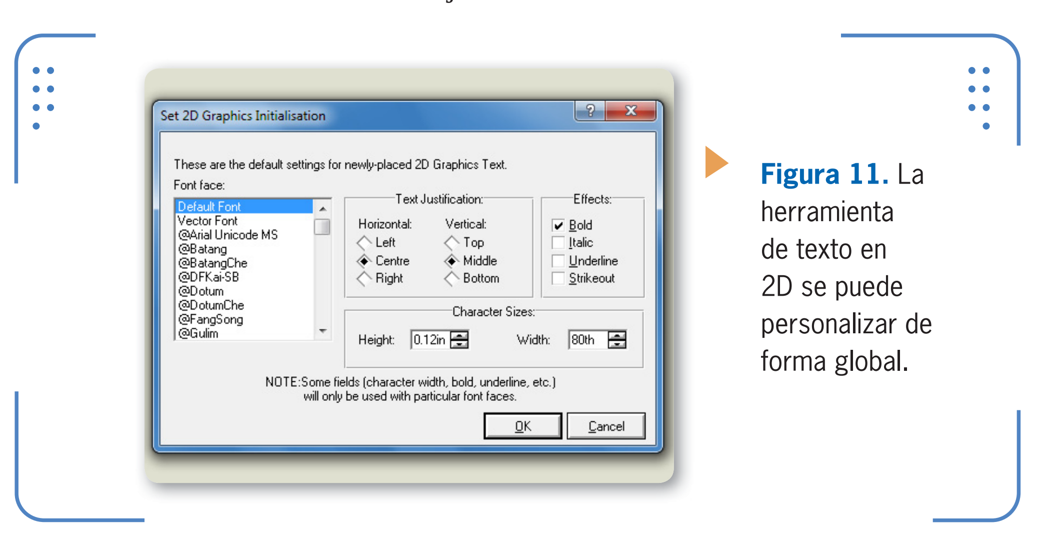


Figura 11. La herramienta de texto en 2D se puede personalizar de forma global.

Es posible elegir el tipo de fuente, los efectos, la justificación, la altura y el ancho del texto predeterminados de la opción de texto 2D. Aunque también, como ya vimos en otros elementos del módulo ISIS, se pueden personalizar los textos de forma local, cada vez que usemos la función de texto en 2D.

Los puntos de unión

Los puntos de unión también pueden ser personalizados, para lo cual vamos al menú **Template** y seleccionamos la opción **Set Junction Dots...**.

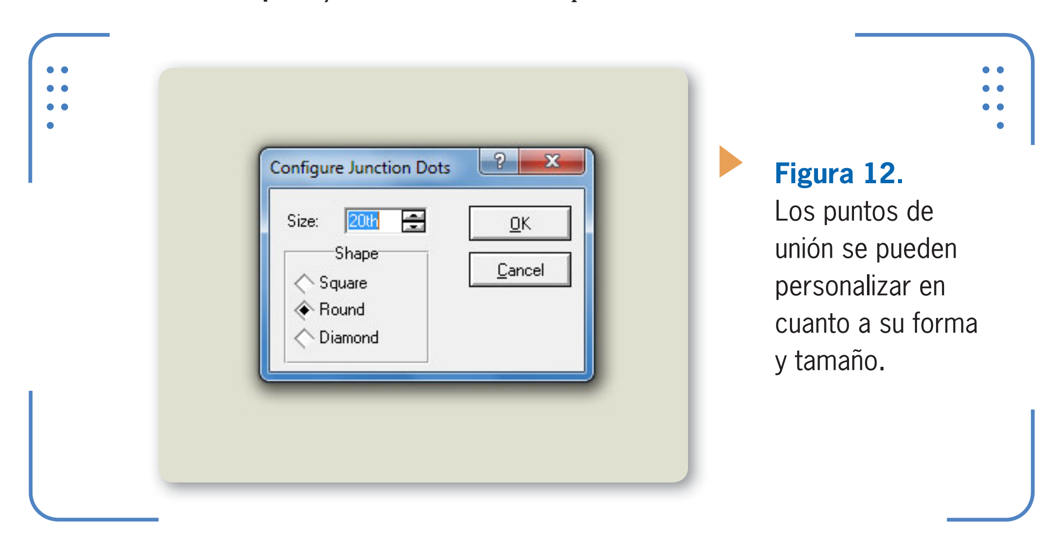


Figura 12. Los puntos de unión se pueden personalizar en cuanto a su forma y tamaño.

Se abrirá **Configure Junctions Dots**, donde elegimos la forma: cuadrados (**Square**), redondos (**Round**) o en diamante (**Diamond**); y el tamaño en **Size**.

Cargar estilos de otro diseño

Al realizar cambios en los estilos globales, tanto en los colores como en los textos, estos solo se aplican para el diseño en el que los realicemos; mientras que si abrimos otro diseño o creamos uno nuevo, estos tendrán los colores por defecto. Cada diseño guarda sus colores personalizados dentro del mismo archivo **.DSN**; por eso, si queremos recuperar un grupo de estilos creado en otro diseño, podemos hacerlo tomándolo del archivo **.DSN** que lo contiene.

CADA DISEÑO GUARDA SUS COLORES PERSONALIZADOS DENTRO DEL ARCHIVO .DSN



Para esto, debemos ir al menú **Template** y seleccionar **Load Styles from Design...**. Al hacerlo, se abrirá la ventana para explorar el sistema y localizar el archivo **.DSN** que buscamos. Cuando lo abrimos, se cargan los estilos de ese diseño, lo que facilita recuperar los estilos personales sin tener que crearlos otra vez.

Plantillas y el estilo por defecto

Al crear un nuevo diseño en ISIS, ya sea si abrimos el programa desde su icono o si presionamos el botón **New File** de la barra de opciones de archivo, siempre tendrá el estilo por defecto; es decir, la combinación de colores con el fondo amarillo, los componentes con líneas de color marrón y relleno amarillo oscuro, el que hemos visto hasta ahora. Este estilo se carga siempre porque está guardado en una plantilla que se usa de manera predefinida en todos los diseños nuevos.

Si hemos realizado cambios en los colores y los textos de un diseño, y al finalizar no fueron de nuestro agrado, podemos revertirlos aplicando de nuevo la plantilla de estilos por defecto. Para lograrlo, simplemente vamos al menú **Template** y seleccionamos **Apply Default Template...**; con esto se carga la plantilla por defecto en el diseño y podemos iniciar de nuevo la personalización, si lo deseamos.

Edición y creación de plantillas de estilos

Las plantillas son útiles para cargar opciones de estilos y fondos para nuevos diseños; por eso, si tenemos un estilo personalizado, podemos guardarlo como una plantilla. Cuando ingresamos al menú **File** y seleccionamos la opción **New Design...**, aparece la ventana **Create New Design**, en la cual se encuentran todas las plantillas disponibles. En las plantillas se guardan los siguientes elementos o configuraciones:

- Todos los estilos globales de colores y textos.
- Todo lo que se encuentre en la hoja global.
- Los componentes que estén en el **Selector de objetos**.



COMPONENTES Y PLANTILLAS

Al crear una plantilla, se guardarán en ella los componentes que estén en ese momento en el **Selector de objetos**. Esto puede resultar muy útil en el caso de componentes que utilicemos muy frecuentemente. Cuando abrimos un nuevo documento con esa plantilla, los componentes ya estarán listos para usarse en el nuevo diseño y nos ahorraremos su búsqueda.

Teniendo esto en cuenta, podemos crear plantillas con colores y textos personalizados, con un cuadro de datos u otros elementos colocados en la hoja global, y los componentes que estén en el **Selector de objetos** en ese momento. Para guardar una plantilla, vamos al menú **File** y pulsamos **Save Design As Template...**; se abrirá el explorador para elegir un nombre y una ubicación. Las plantillas tienen la extensión **.DTF** y se almacenan en la carpeta **TEMPLATES**, que está dentro de la carpeta de instalación de Proteus.

La plantilla por defecto tiene el nombre **DEFAULT.DTF**. Podemos modificarla y sobrescribirla, con lo cual la plantilla por defecto será la que hayamos definido con todos los cambios hechos por nosotros. Es recomendable hacer una copia de respaldo antes de modificar la plantilla **DEFAULT.DTF**, en caso de querer recuperarla en el futuro.

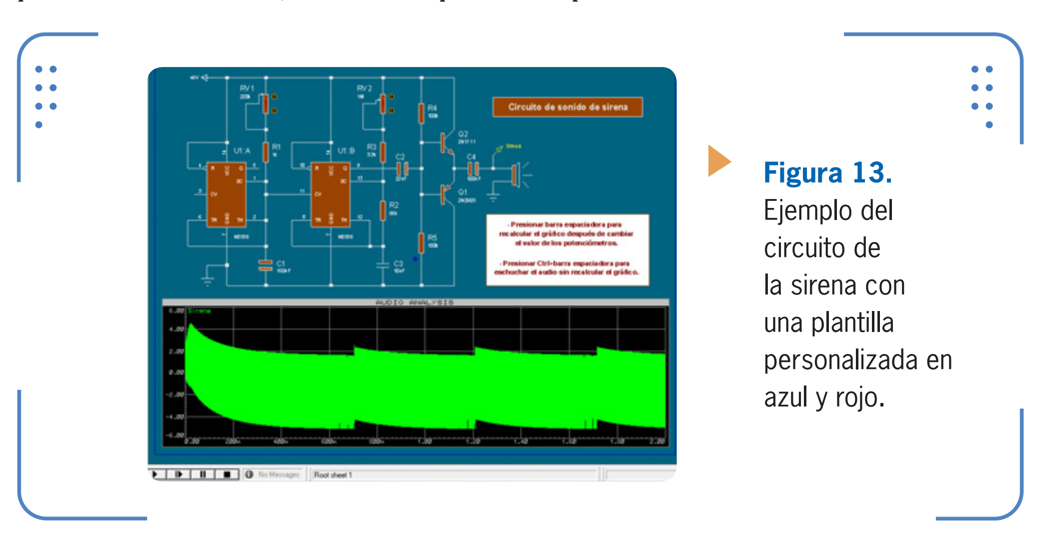


Figura 13. Ejemplo del circuito de la sirena con una plantilla personalizada en azul y rojo.



RESUMEN

En este capítulo vimos en detalle cómo podemos personalizar la interfaz de ISIS para adaptarla a nuestras necesidades o gustos, cambiando algunos elementos de lugar o modificando la forma en que los diseños se muestran en la ventana de edición al variar los estilos de los colores y los textos. Aunque la configuración por defecto en ISIS está diseñada estratégicamente para facilidad y comodidad de uso, es bueno saber que podemos cambiar algunos elementos y adaptarlos a nuestro gusto o necesidad. Ahora sabemos cómo hacerlo.

Actividades

TEST DE AUTOEVALUACIÓN

1 ¿Cómo se cambia de lugar o posición una barra de herramientas?

2 ¿Cuál es el procedimiento para ocultar la barra de herramientas de edición?

3 ¿Cómo podemos hacer que el **Selector de objetos** se oculte automáticamente?

4 ¿Qué es la hoja global?

5 ¿Desde dónde debemos editar un cuadro de datos en las plantillas de Proteus?

6 ¿Qué es un estilo global?

7 ¿Qué es un estilo local?

8 ¿Qué es una plantilla?

9 ¿Qué configuraciones se guardan en las plantillas?

10 ¿Cómo se guarda una plantilla?

ACTIVIDADES PRÁCTICAS

1 Tome alguno de los diseños de los capítulos anteriores (excepto la fuente variable) y cree un cuadro de datos personalizado en él, con el tamaño, la forma, los colores y textos que usted quiera.

2 Haga un respaldo de la plantilla **DEFAULT.DTF**.

3 Abra ISIS desde su icono, vaya al menú **Template/Set Design Defaults...** desmarque la opción **Show Hidden Text?** y haga clic en **OK**. Guarde la plantilla como default; con esto, los textos ocultos no se mostrarán más en el futuro.

4 Cree un grupo de estilos personalizados con los colores y textos de su agrado, e incluya un cuadro de datos si lo desea.

5 Guarde los estilos personalizados como una plantilla nueva para usarla en otros diseños.



Opciones de simulación y corrección de errores

Cuando dibujamos un circuito en ISIS e intentamos simularlo, puede ser frustrante obtener errores que no permiten que el proceso se realice y nos impiden comprobar si el circuito funciona. Algunos de estos errores se deben a que el circuito no está bien dibujado o diseñado, y otros, a que el simulador no puede resolverlo. En este capítulo veremos las opciones de simulación y cómo corregir los errores.



↗↗↗



Operación básica de ProSPICE

Para entender algunos de los errores que pueden presentarse durante una simulación, es bueno conocer cómo funciona el núcleo del simulador. Como sabemos, la simulación analógica en Proteus es realizada por el programa **ProSPICE**. Intentaremos comprender cómo se lleva a cabo una simulación sin introducirnos en las matemáticas complejas que esto involucra; solo nos interesa tener una idea del procedimiento con el que se efectúa el cálculo de una simulación, principalmente, en los circuitos analógicos.

Para comenzar, en un circuito analógico lo que se pretende conocer, básicamente, son los voltajes y las corrientes existentes en cada punto del circuito. Tomemos como ejemplo un circuito puramente resistivo.

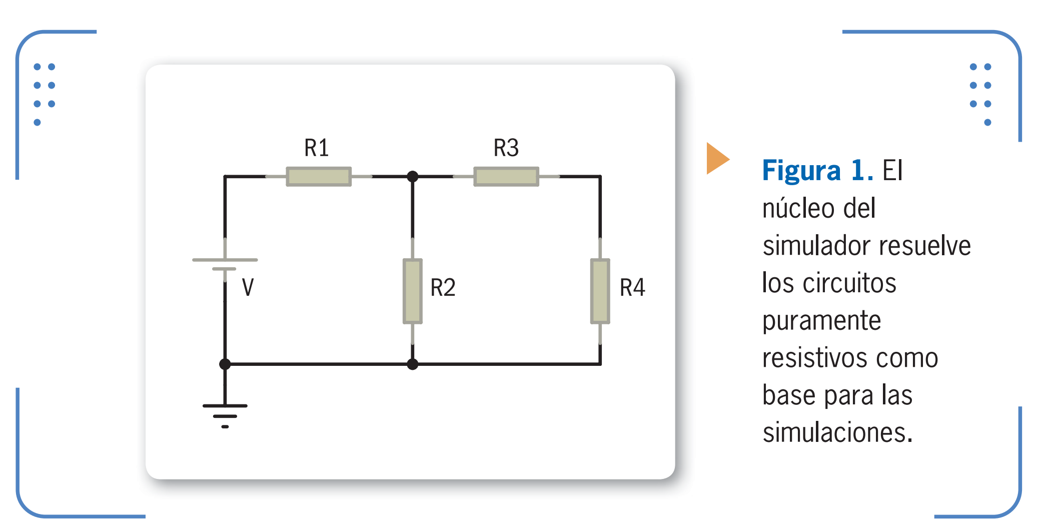


Figura 1. El núcleo del simulador resuelve los circuitos puramente resistivos como base para las simulaciones.



MÉTODOS NUMÉRICOS Y COMPUTADORAS

Los métodos numéricos requieren de una enorme cantidad de iteraciones para resolver las ecuaciones, y hacerlo de forma manual es imposible en algunos casos. Es aquí donde cobra importancia la aparición de las computadoras, que pueden procesar una enorme cantidad de cálculos en muy poco tiempo.

Para este tipo de circuito, ProSPICE genera una **matriz** usando la ley de las corrientes de Kirchoff (**LCK**) y la resuelve con métodos algebraicos como la eliminación gaussiana; así se pueden calcular los voltajes y las corrientes en cada nodo del circuito.

Elementos no lineales

Cuando el circuito contiene elementos **no lineales**, el asunto se complica un poco. Para entenderlo más claramente, por ejemplo, tomemos un circuito donde tenemos un diodo.

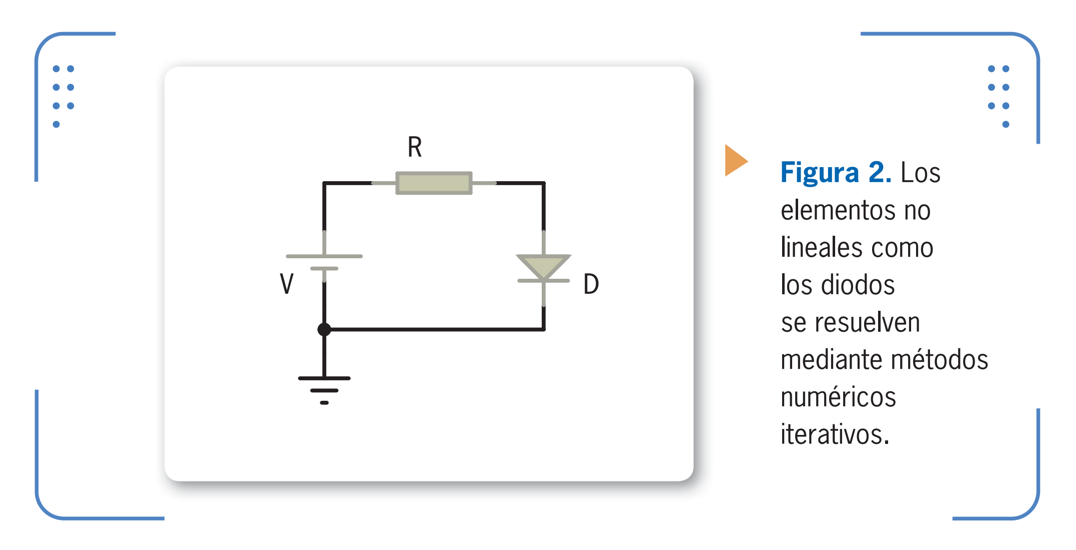


Figura 2. Los elementos no lineales como los diodos se resuelven mediante métodos numéricos iterativos.

La ecuación que representa la relación entre el voltaje y la corriente en el diodo hace imposible usar métodos algebraicos simples para resolver el circuito mediante **LCK**, porque no es lineal. Es aquí donde se deben usar **métodos numéricos**, que consisten en proponer un punto de operación inicial para el circuito y, a partir de él, generar **iteraciones**, es decir, repeticiones de cálculos para ir acercándose a la solución.

En el ejemplo que vemos en la **Figura 3**, la solución está dada por la intersección de la recta de carga y la curva del diodo. En la aproximación inicial se calcula la derivada de la curva para obtener una ecuación lineal que puede resolverse, pero no es la solución buscada, así que se hace otro cálculo o iteración con el nuevo punto de operación obtenido. Así, con cada iteración se acerca cada vez más a la solución real, por lo que se dice que el método **converge**. El número

de iteraciones se detiene cuando los valores obtenidos varían muy poco de una iteración a otra; en este punto se puede considerar que se ha llegado al resultado. De esta forma, se ha **linealizado** el comportamiento del diodo, a partir del cual ya se puede resolver el circuito usando LCK, y obtener las corrientes y voltajes en él.

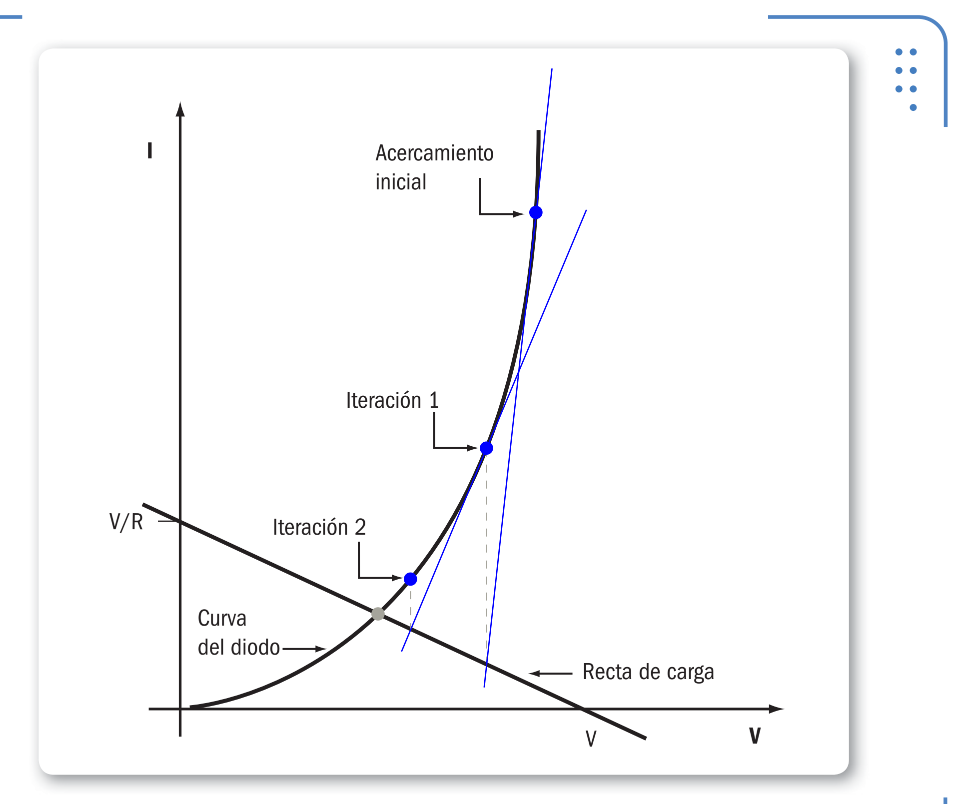


Figura 3. Se usa el método llamado de **Newton-Rapshon** para resolver las ecuaciones de elementos no lineales.





MODELOS NUEVOS EN LA ÚLTIMA VERSIÓN

En **www.labcenter.com/vmodels/peripherals.cfm** encontramos una lista de los modelos avanzados que contiene Proteus, principalmente, circuitos integrados y periféricos, que pueden simularse. Los modelos agregados a la última versión están resaltados, para poder acceder a ellos de manera inmediata.

Componentes reactivos

Una complicación más se presenta con los componentes reactivos: inductores y capacitores. La dificultad radica en que las expresiones que describen su comportamiento son **ecuaciones diferenciales**. Aquí se usan otra vez métodos de integración numéricos para resolver los circuitos que los contienen. El primer método es el trapezoidal.

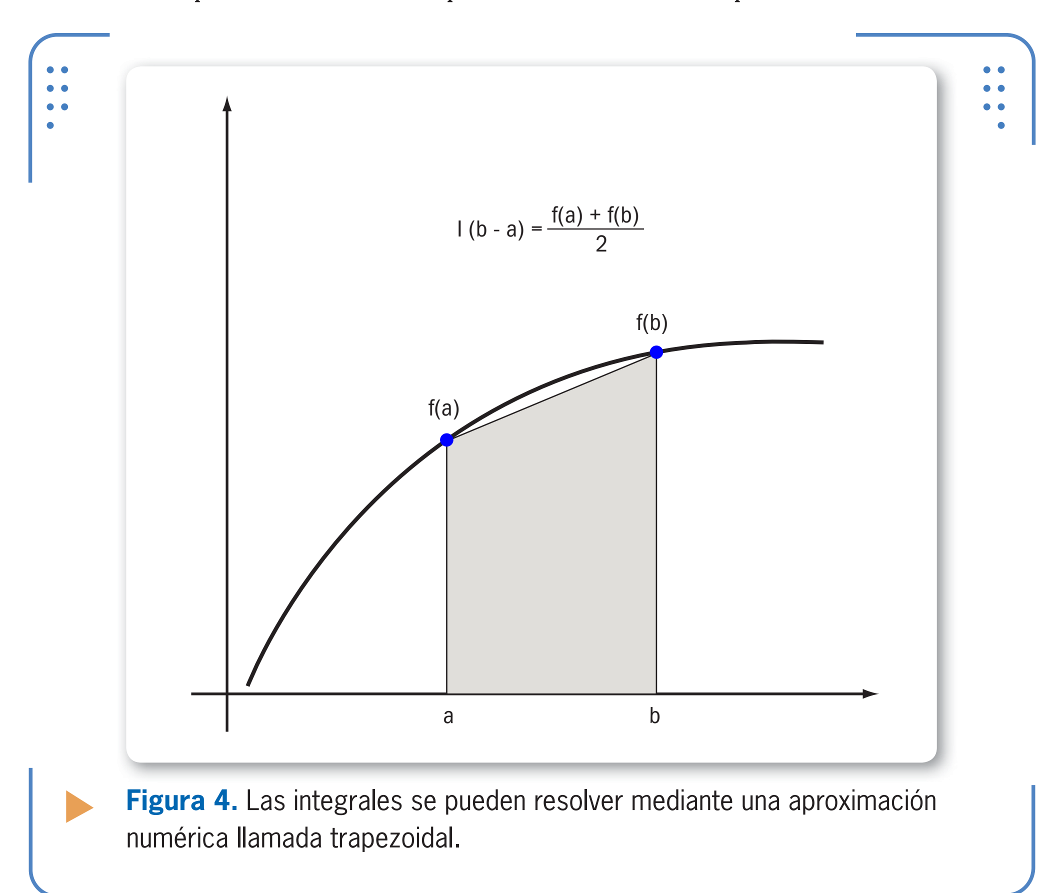


Figura 4. Las integrales se pueden resolver mediante una aproximación numérica llamada trapezoidal.

El segundo método es el de **Gear** o BDF (*Backward Differentiation Formula*). Este proceso es también iterativo y, mediante la simplificación de las ecuaciones, busca encontrar el punto de operación usando el voltaje en un tiempo previo, la corriente actual y un intervalo de tiempo llamado **timestep**. De esta forma, es posible hallar un equivalente del capacitor o inductor como un circuito resistivo, que, como ya vimos, se puede resolver con álgebra simple.



Las opciones de simulación

En el menú **System/Set Simulator Options...** tenemos una ventana donde es posible cambiar las opciones internas de simulación, es decir, los parámetros que usarán **ProSPICE** y **DSIM**. Estos determinan la exactitud, la velocidad y la capacidad de resolver las simulaciones.

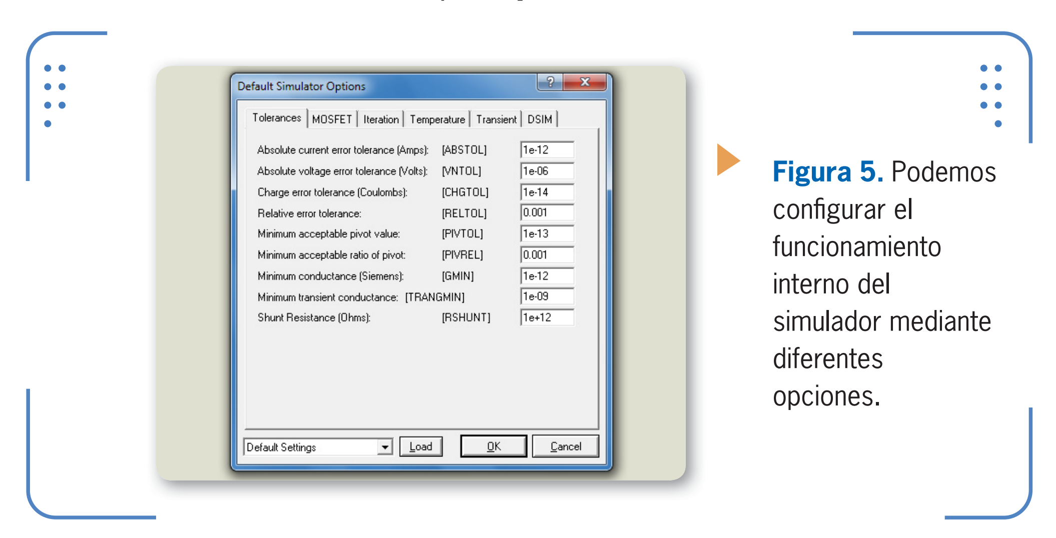


Figura 5. Podemos configurar el funcionamiento interno del simulador mediante diferentes opciones.

En la pestaña **Tolerances** encontramos opciones de tolerancia para diversos parámetros de funcionamiento interno de ProSPICE. Podemos destacar: la tolerancia de corriente (**ABSTOL**), de voltaje (**VNTOL**) y de carga (**CHGTOL**), que definen las cantidades respectivas a las cuales se debe llegar para que se considere que la solución de un circuito converge; las cantidades menores a estos valores son ignoradas por el simulador. La tolerancia relativa de error (**RELTOL**) establece el límite para las cantidades anteriores; es decir, cuando entre una iteración y




MATEMÁTICA PURA

El motor de Proteus, que es básicamente ProSPICE, simula todos los circuitos modelando de forma matemática los componentes y generando los resultados a través de la resolución de las ecuaciones del circuito. Por eso podemos decir que, en esencia, ProSPICE es un programa para resolver ecuaciones.

otra se tiene este valor relativo, se considera que se ha llegado a la solución. El valor por defecto es 0,001, es decir, el 0,1%.

El parámetro **GMIN** determina la conductancia mínima permitida por ProSPICE. Este valor define la corriente de fuga para las uniones de semiconductores polarizadas inversamente y para otros puntos teóricos de impedancia infinita. Reducir este valor suele ayudar a los circuitos que fallan al intentar simularse, pero al hacerlo, se puede perder exactitud. Valores mayores de 1e-9 tienden a dar resultados dudosos.

EL PARÁMETRO GMIN DETERMINA LA CONDUCTANCIA MÍNIMA PERMITIDA POR PROSPICE

Mediante la pestaña **MOSFET** es posible cambiar algunos parámetros físicos del funcionamiento de los transistores MOSFET, como, por ejemplo, el área de difusión. Debemos tener en cuenta que no es necesario mover estos valores a menos que estemos diseñando circuitos integrados, que es en donde tienen una utilidad real.

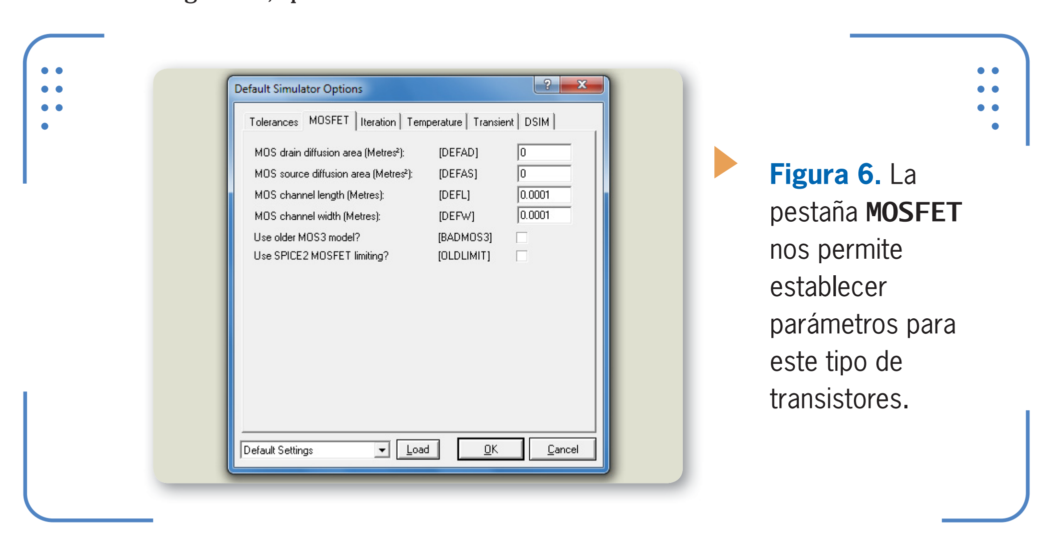


Figura 6. La pestaña **MOSFET** nos permite establecer parámetros para este tipo de transistores.

En la pestaña **Iteration** hay opciones para el control de los cálculos. Se destaca el método de integración (**METHOD**), que puede ser TRAPEZOIDAL o GEAR (es más estable y es el predefinido). **GMINSTEPS** determina el número de pasos en el modo **GMIN stepping**. En este modo, ProSPICE aumenta el valor **GMIN** en las simulaciones cuando los cálculos no convergen; al encontrar un valor en el que todo converge,

lo toma para resolver el circuito y regresa el valor de **GMIN** a su número inicial para continuar y asegurar que la simulación funcione.

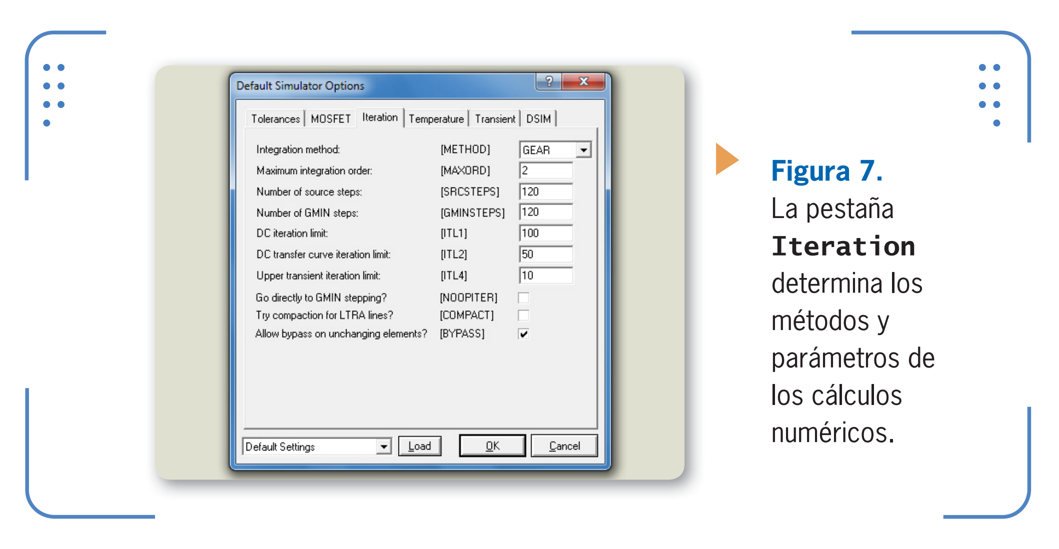


Figura 7. La pestaña **Iteration** determina los métodos y parámetros de los cálculos numéricos.

Los campos para el límite de iteraciones (**ITL1**, **ITL2** e **ITL4**) establecen el número máximo de iteraciones de los cálculos en DC, de curvas de transferencia o de análisis temporal, respectivamente.

En la pestaña **Temperature** hallamos dos parámetros: la temperatura a la que está el circuito (**TEMP**) y la temperatura nominal que se usa para calcular parámetros dependientes de la temperatura en los componentes (**TNOM**), por ejemplo, en las uniones de semiconductores.

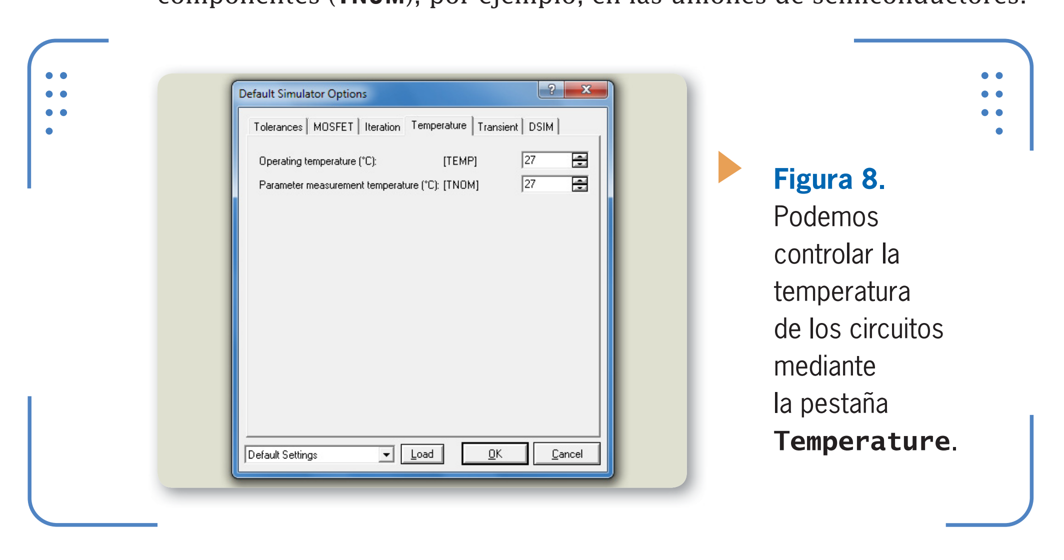


Figura 8. Podemos controlar la temperatura de los circuitos mediante la pestaña **Temperature**.



Errores al simular

Al intentar simular un circuito, podemos encontrarnos con errores que hacen que el proceso no funcione o, al menos, no como se espera. En general, estos problemas son de diseño y no debidos a fallas en Proteus; algunos otros errores ocasionarán que la simulación se detenga al estar corriendo o ni siquiera sea capaz de iniciarse. A continuación, veremos algunos casos en particular y el modo de solucionarlos.

Carga del procesador

Como sabemos, los circuitos muy complejos pueden generar una carga importante al procesador y no correr en tiempo real, esto no es en realidad un error, pero sí, un detalle que impide ver la simulación tal como es. Para estos casos, daremos algunos consejos que permiten simplificar los circuitos y lograr una simulación más fluida.

En primer lugar, en algunos diseños, es posible dividir los circuitos. Por ejemplo, al simular un amplificador, podemos alimentarlo con las fuentes de ISIS y no tratar de simular la fuente en el mismo circuito. Al eliminar la fuente de alimentación, la carga se aligerará.

AL INTENTAR SIMULAR UN CIRCUITO, PODEMOS ENCONTRARNOS CON DISTINTOS ERRORES



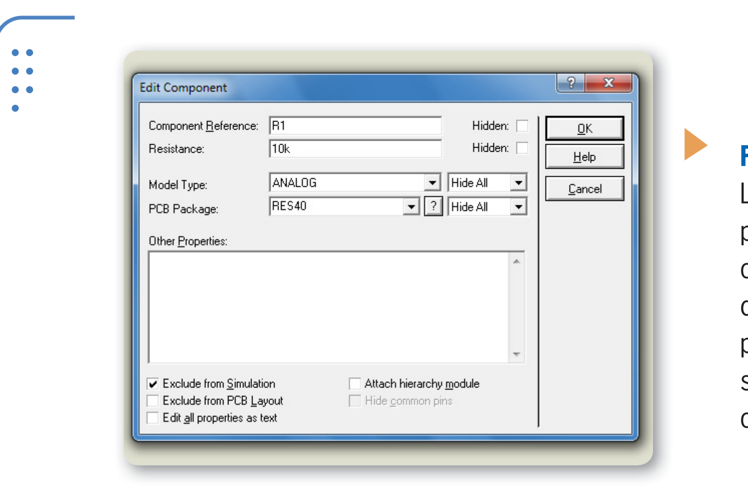
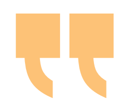


Figura 9. La opción para excluir componentes de la simulación puede ayudar a simplificar los circuitos.

PODEMOS EXCLUIR LOS COMPONENTES NO CRÍTICOS PARA REDUCIR LA CARGA DEL PROCESADOR



También podemos excluir los componentes no críticos de la simulación, por ejemplo, todos los capacitores de desacople.

Al marcar la opción de exclusión en sus propiedades (ventana **Edit Component**), hacemos como si los componentes no estuvieran presentes en el circuito. Esto puede aligerar la carga al simular, aunque en algunos casos suele ser difícil encontrar componentes que realmente podamos excluir sin afectar el funcionamiento del circuito.

Errores de conexión

Algunos de los errores se deben a conexiones faltantes o equivocadas. En estos casos, el circuito tiende a funcionar de manera incorrecta. La única solución que nos queda es revisar todas las conexiones y asegurarnos de que se han realizado de forma adecuada. Para hacerlo, podemos auxiliarnos con el **verificador de conexiones eléctricas** que estudiamos en el **Capítulo 7**. Debemos controlar que todas las terminales y los pines de componentes estén realmente conectados al circuito y en los lugares correctos.

Las referencias de componentes

Dentro de cualquier circuito, los componentes deben tener una referencia numerada; por ejemplo, los resistores tienen que llevar las referencias: R1, R2, R3, etcétera. Si dos o más componentes tienen la misma referencia, al intentar simular obtendremos el siguiente error en el informe de simulación: **Duplicate part reference: Xn [Xn]** en color rojo, donde **X** representa el tipo de componente, y **n**, el número.



CARGA DE LA CPU Y LA ANIMACIÓN

Si tenemos activadas las opciones de animación para mostrar voltajes con colores, niveles lógicos o corrientes, debemos tomar en cuenta que esto genera un poco de carga extra de procesamiento. Para aligerar las simulaciones podemos desactivar todas las opciones de animación en circuitos complejos.

Figura 10. Ejemplo de un circuito con componentes que tienen referencias idénticas y generará un error.

En el caso de que necesitáramos simular el circuito que se muestra en la **Figura 10**, la simulación no podría comenzar y obtendríamos un error porque, como podemos observar en el diseño, la referencia de los resistores marcados como **R2** se encuentra duplicada.

Entonces, en el informe de simulación aparecerá precisamente un error del tipo **Duplicate part reference**, que nos informará que dos resistores poseen la misma referencia.



MÉTODOS ANALÍTICOS CONTRA NUMÉRICOS

Los métodos analíticos se encargan de resolver problemas matemáticos de forma directa, pero algunas veces esto no es posible, y es aquí donde los métodos numéricos entran en escena. Estos métodos resuelven los problemas mediante cálculos aritméticos simples, generalmente, con gran cantidad de repeticiones o iteraciones, para acercarse poco a poco a la solución buscada. El núcleo del simulador usa métodos numéricos para resolver los circuitos.

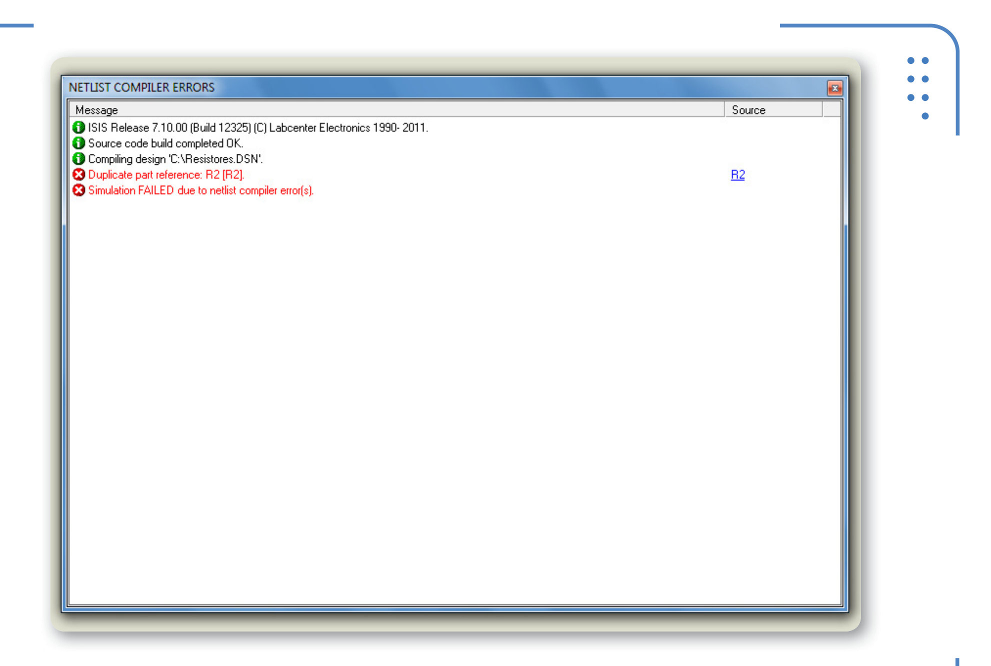


Figura 11. El informe de simulación marca las referencias duplicadas y permite ir al componente para corregirlo.

Entonces, el informe muestra cuál es la referencia que se encuentra duplicada. En la parte derecha, podemos ver una columna llamada **Source**, donde se indica **R2** en azul; si hacemos clic en esta referencia, el informe de simulación se cerrará, el componente se centrará en la ventana y se marcará con rojo. De esta manera, podremos saber fácilmente cuál es la referencia que está repetida, si tener que revisar una por una en el diseño del circuito.



EL ANOTADOR GLOBAL Y LAS HOJAS

Dentro de las propiedades de las hojas encontramos una interesante opción denominada **Annotator Init**, cuya función es indicarle al **anotador global** cuál es el inicio de la numeración de las referencias para esa hoja en particular. Por ejemplo, es posible comenzar la numeración en una hoja a partir del número 10, y los resistores en ellas se numerarán como R10, R11, R12, etcétera; o podemos iniciar la numeración en 20, y los resistores se numerarán como R20, R21, R22, etc.

Figura 12. Al hacer clic en R2 en el informe de simulación, se resaltará el componente con referencia duplicada.

El anotador global

Si obtenemos el error de referencias repetidas, es posible editarlas manualmente, pero en circuitos grandes suele ser difícil saber qué etiqueta colocar para no reiterar la de otro componente. Para solucionar este problema, es mejor hacerlo de forma automática con el **anotador global**. Accedemos a él desde el menú **Tools/Global Annotator...**. En la ventana **Annotator** configuraremos cómo se realizará la anotación automática. En la **Guía visual** explicaremos las opciones.



ERRORES DE DISEÑO

Si un circuito no funciona o no hace lo esperado, la mayoría de las veces se debe a errores de diseño o a conexiones incorrectas. Antes de pensar que el simulador no sirve, es bueno verificar si el diseño es correcto y si dibujamos bien el circuito. El simulador no puede decirnos si el circuito está mal diseñado.

GUÍA VISUAL ■ ANOTADOR GLOBAL

01 **INITIAL COUNT:** define el inicio de la cuenta para las referencias.

02 **INCREMENTAL:** solo se editarán las referencias de los componentes que todavía no las tengan.

03 **TOTAL:** en este modo se editarán las referencias de todos los componentes.

04 **CURRENT SHEET:** se editarán las referencias solo en la hoja actual.

05 **WHOLE DESIGN:** con esta opción se editarán las referencias de todo el diseño.

La anotación en tiempo real

La opción **Incremental** del anotador global funciona sobre los componentes que no tienen referencia numerada. Al dibujar un diagrama, los componentes van tomando la numeración de sus referencias de forma automática; esto es lo que se llama **anotación en tiempo real** y se puede desactivar desde el menú **Tools/Real Time**

Annotation. Al hacerlo, en lugar de numerarse automáticamente, las referencias serán marcadas con un signo de interrogación, por ejemplo, todos los capacitores se marcarán como **C?**. Esto se puede hacer para dejar la numeración pendiente y, al final, cuando el diagrama esté completo, usar el anotador global para asignar la numeración.

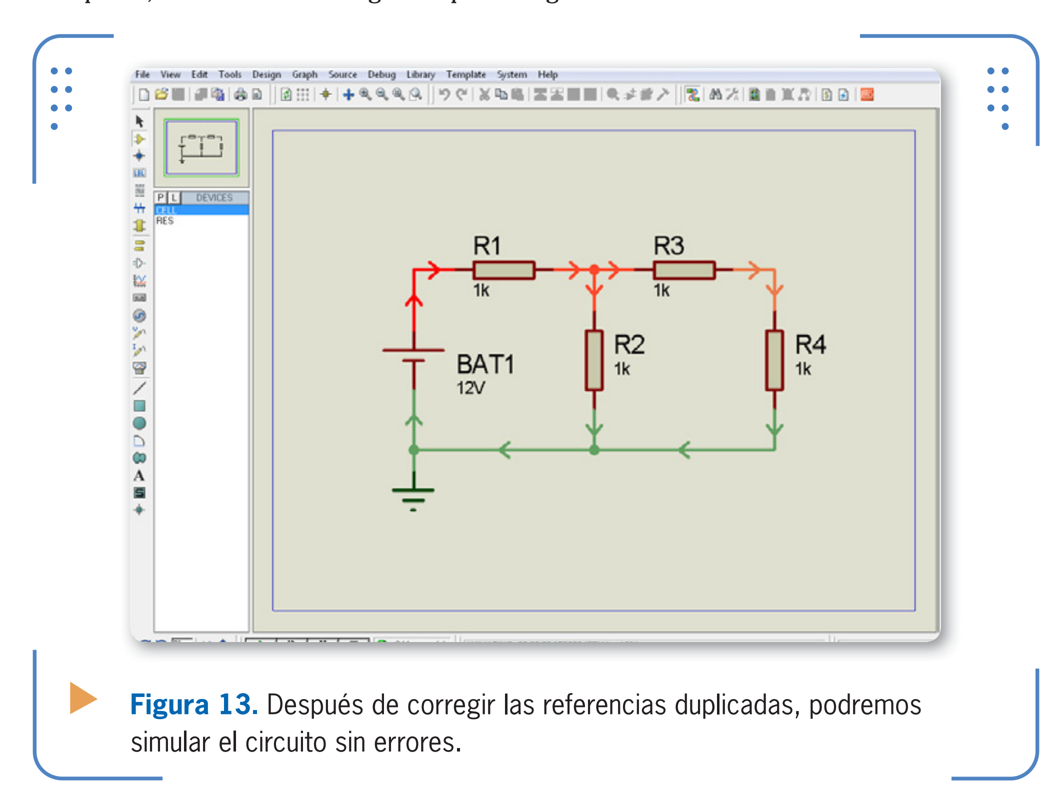


Figura 13. Después de corregir las referencias duplicadas, podremos simular el circuito sin errores.

Por supuesto que, si intentamos simular un circuito que carezca de referencias numeradas, obtendremos el mismo error, ya que todas las referencias se tomarán como duplicadas. El anotador global puede numerar de forma automática las referencias no definidas, y además, corregir todas las referencias duplicadas.

Error de componentes sin modelo

Si intentamos simular un circuito que contiene componentes sin modelo de simulación, obtendremos el siguiente error: **No model specified for: Xn**, donde **X** es el componente, y **n**, su número de

referencia. Como estudiamos antes, los componentes que no cuentan con modelo no pueden simularse, ya que al intentarlo, el simulador simplemente no sabe qué hacer con ese componente.

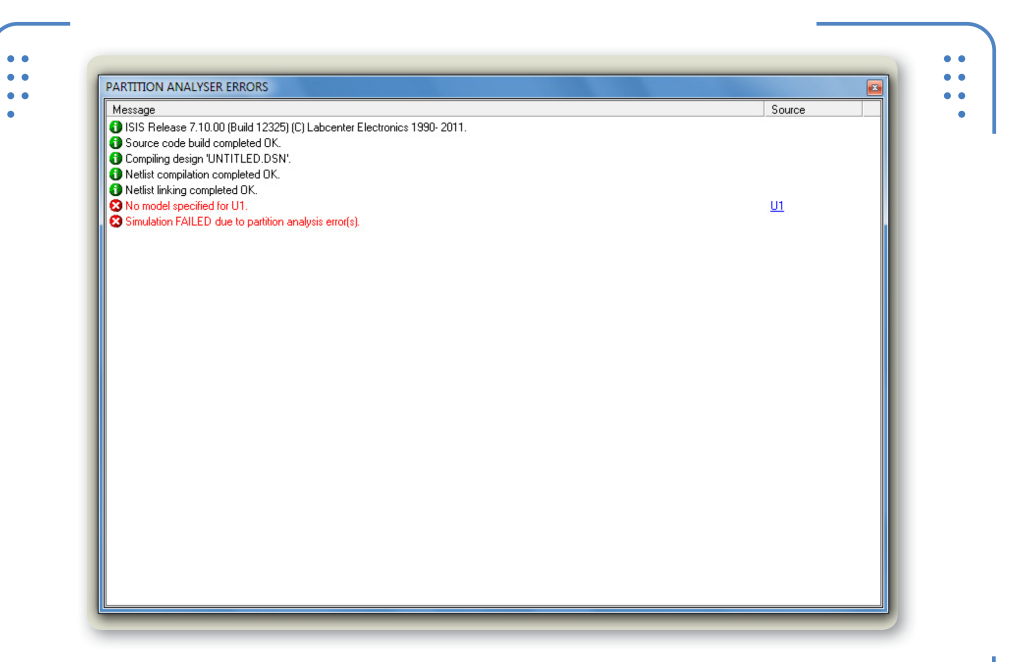


Figura 14. Como podemos intuir, es imposible simular un circuito que contiene componentes sin modelo de simulación.

Para corregir este error podemos hacer lo siguiente: si el componente no es esencial para la simulación (por ejemplo, un conector), podemos borrarlo o excluirlo de la simulación; si el componente es parte fundamental del circuito, podemos buscar un **reemplazo**, por ejemplo, sustituir un amplificador operacional sin modelo por alguno similar que sí lo tenga.

Errores de convergencia

El último grupo de errores que analizaremos son los errores internos del simulador; es decir, cuando por alguna razón el simulador no puede encontrar una solución para las ecuaciones del circuito y, por lo tanto, no puede simularlo. Los errores de este tipo se explican a continuación.

Singular matrix

Este error es causado cuando una matriz con la que se intenta resolver un circuito tiene más incógnitas que ecuaciones y, entonces, no puede calcularse. Generalmente, es debido a un circuito mal diseñado o con conexiones erróneas o faltantes, o también a que parte del circuito no está bien conectado a tierra, por lo que no se pueden resolver los voltajes, ya que no hay referencia.

Too many iterations without convergence

Esto significa que los cálculos de un circuito o parte de él son inestables, y no se puede llegar a una solución aun después de haber intentado muchas iteraciones.

Figura 15. En este informe tenemos un ejemplo de un error de convergencia que impide que la simulación se lleve a cabo.

Timestep too small

Cuando ocurre este problema, el circuito entra en un estado donde, al avanzar el tiempo en intervalos muy cortos (típicamente 1e-18

segundos), no se producen cambios aceptables en los voltajes, por lo que la simulación se interrumpe. Este error suele estar producido por modelos de transistores o diodos que no funcionan como corresponde.



Corregir errores de convergencia

Cuando un circuito no puede simularse debido a inconvenientes de convergencia, generalmente estos errores no están aislados; pueden presentarse en conjunto o, incluso, en combinación con problemas en el valor de **GMIN**, ya que también se involucran parámetros propios de las uniones en los semiconductores.

Un buen consejo para solucionar problemas de convergencia es sustituir componentes, principalmente, los diodos o transistores. Si estamos usando diodos o transistores **genéricos**, podemos intentar cambiarlos por reemplazos específicos y verificar si los problemas cesan o no. En el caso de amplificadores operacionales, es posible usar un modelo ideal, que encontramos en la categoría **Operation Amplifiers/ Ideal** en la ventana **Pick Devices**. Este es más fácil de simular que los amplificadores operacionales específicos.

Configurar las opciones de simulación

En ocasiones, configurar las opciones de simulación suele ayudar con circuitos que no simulan o que funcionan de manera incorrecta. En general, podemos tratar de cambiar los valores de algunos parámetros:



CREAR COMPONENTES SUSTITUTOS

En algunos casos podemos diseñar un componente sustituto para un circuito integrado. Por ejemplo, si necesitamos un registro de aproximaciones sucesivas, el módulo ISIS no cuenta con ninguno (o al menos no que tenga modelo de simulación), por lo que podemos diseñar uno utilizando componentes discretos (flip flops, compuertas, etc.) y usarlo como un subcircuito.

- **GMIN**: aumentar su valor pero sin pasar de 1e-09, ya que más allá obtendremos resultados incongruentes.
- **RELTOL**: incrementar su valor para elevar el rango de tolerancia y ver si esto logra que la simulación funcione.
- **Iteration**: en esta pestaña podemos intentar aumentar el número de iteraciones en **ITL1**, **ITL2** o **ITL4** para ver si con un número mayor se logra llegar a las soluciones. También es posible usar el método de integración **TRAPEZOIDAL**, para ver si obtenemos una corrección.

Ajustes predefinidos de las opciones de simulación

Modificar las opciones de simulación puede ser de gran utilidad en algunos casos, aunque también suele ser difícil saber qué hemos alterado, y podemos comenzar a obtener resultados falsos en la simulaciones o empeorar todavía más los errores. Para resolver esta situación, es posible usar las alternativas que figuran en la parte inferior de la ventana de las opciones de simulación.

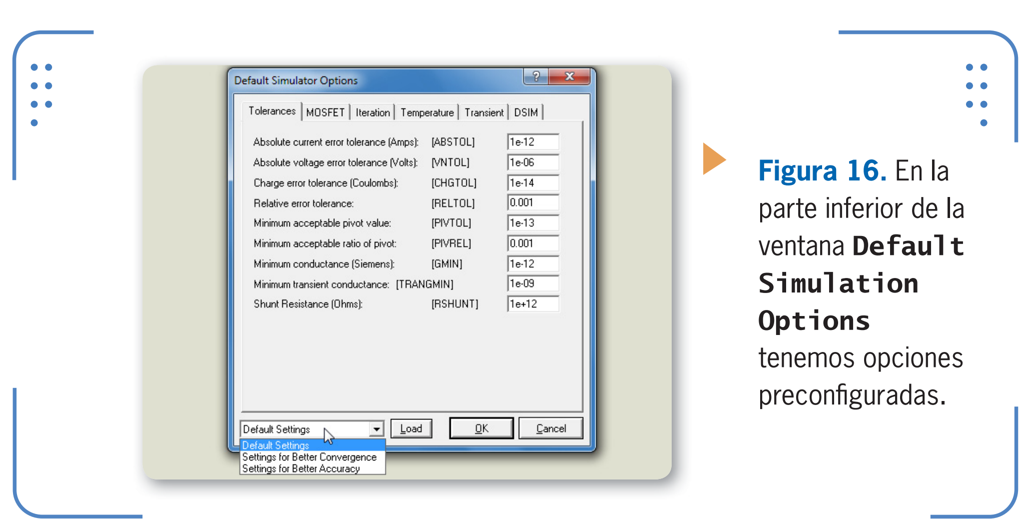


Figura 16. En la parte inferior de la ventana **Default Simulation Options** tenemos opciones preconfiguradas.

En la lista encontramos tres opciones:

- **Default Settings**
- **Settings for Better Convergence**
- **Settings for Better Accuracy**

Podemos valernos de ellas para realizar una configuración de forma rápida. La primera cargará las opciones por defecto; al presionar el botón **Load**, veremos que las opciones se ajustan, lo que sirve para recuperar las opciones iniciales después de haber hecho cambios sin buenos resultados.

La segunda opción cargará en todas las opciones de simulación valores que están optimizados para obtener una mejor convergencia en los cálculos. Podemos intentar esta configuración en circuitos que interrumpen por alguna causa la simulación, y ver si con esto se corrigen.

La tercera opción carga una configuración para obtener la mejor exactitud posible, sacrificando un poco la velocidad y la posibilidad de convergencia. Podemos intentar esta opción en circuitos que se comportan de forma errática y sospechamos que no están dando resultados correctos o estables.

MODIFICAR LAS OPCIONES DE SIMULACIÓN PUEDE SER UN PROBLEMA EN ALGUNOS CASOS

Opciones de simulación en diferentes lugares

Es posible acceder a las opciones de simulación desde tres lugares distintos, por esta razón es muy importante tomar en cuenta las diferencias entre cada uno de ellos:

- Desde el menú **System/Set Simulator Options...**, la configuración será guardada de forma global; es decir, se usará en todas las simulaciones, incluyendo los diseños que abramos en el futuro. En este caso, el título de la ventana será **Default Simulator Options**.





EL ANOTADOR GLOBAL

Es posible copiar las partes o elementos que necesitan estar duplicados en un circuito usando la opción de pegar al Portapapeles, o también podemos copiar un circuito completo de otro diseño o de varios diseños. Las referencias suelen estar duplicadas al hacerlo de esta manera, pero el anotador global las corregirá por nosotros de una manera sumamente fácil.



- Desde la ventana para configurar las opciones de animación (**System/ Set animation options...**) con el botón **SPICE Options**; la configuración solo será guardada y usada en ese diseño en particular. El título en este caso será **Interactive Simulator Options**.
- Desde las propiedades de un gráfico de simulación con el botón **SPICE Options**; la configuración solo se aplicará para ese gráfico en particular. En este caso, la ventana se llamará **Simulator Options**.



Condiciones iniciales

Es posible definir las condiciones iniciales para una red de conexiones o nodo al momento de comenzar una simulación, proponiendo un voltaje inicial en ella. Para hacerlo, debemos usar la opción **IC=X** (IC significa **Initial Condition**) como una etiqueta de alguna línea de conexión, donde la **X** representa el voltaje deseado.

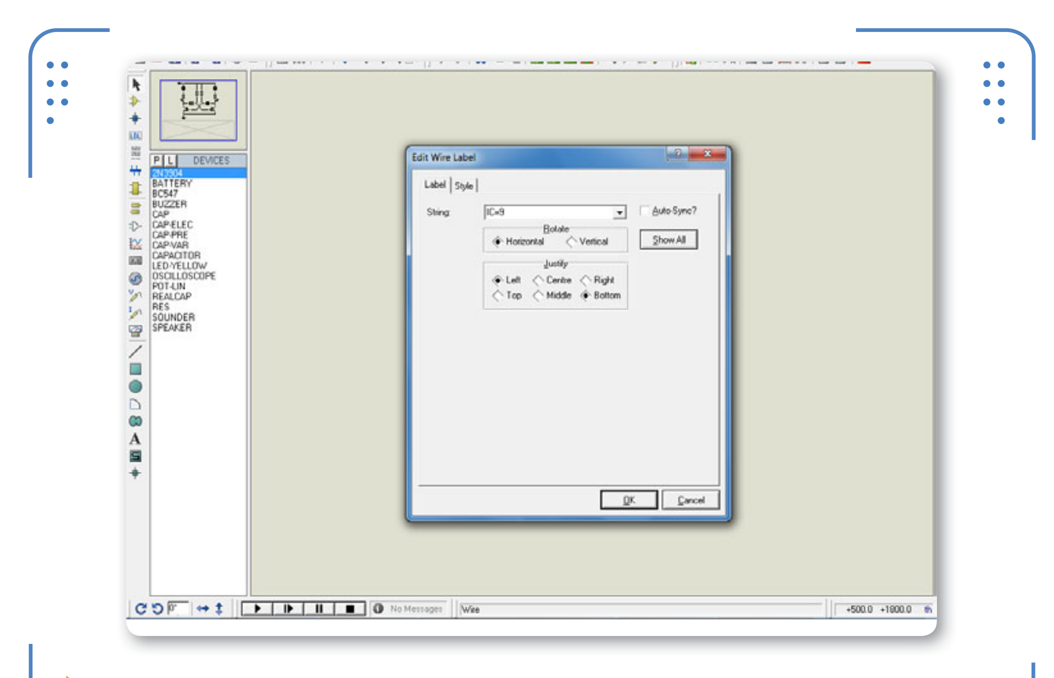


Figura 17. Las etiquetas **IC=X** en las líneas de conexión definen voltajes iniciales para un nodo del circuito, en este ejemplo, **9 Volts**.

Veamos un ejemplo de cómo establecer una condición inicial, para lo cual tomaremos un flip flop con transistores.

Este es un circuito muy conocido, y todos sabemos que realmente funciona de manera correcta; sin embargo, al simularlo, tal vez no arranque o tarde mucho en hacerlo, ya que en la simulación todos los parámetros del circuito son simétricos.

Para lograr que el circuito arranque de manera correcta, podemos hacerlo con una **condición inicial**.

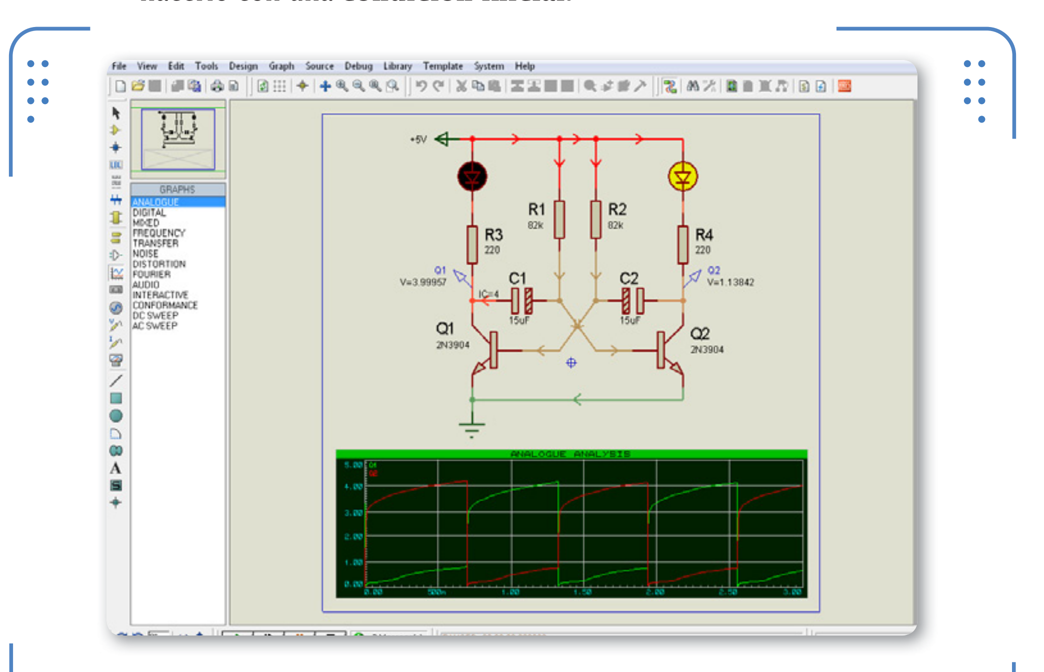


Figura 18. Podemos definir condiciones iniciales en los puntos necesarios para favorecer el arranque de la simulación.



CONDICIONES INICIALES DIGITALES

Para los circuitos que son puramente digitales se pueden establecer condiciones iniciales con la expresión **BS=X**, donde **BS** significa **Boot State** y **X** representa el estado lógico deseado: 0, 1, H, L, HIGH, LOW, SHI, WHI, SLO, WLO, o FLT. Para circuitos mixtos, es posible establecer un voltaje con **IC**, que se propagará como un estado lógico a los elementos digitales.

Colocamos una etiqueta en la línea que conecta el capacitor **C1** con el colector de **Q1** y escribimos **IC=4**, con lo que el estado inicial en esa red o nodo será de **4 Volts**. Podemos descargar el archivo **FF.dsn** para ver este ejemplo. Si iniciamos la simulación presionando el botón **Pause** o **Step**, observaremos que en la red donde colocamos la etiqueta tenemos un voltaje de 4 Volts, y esto permite que el circuito arranque de inmediato.

La condición inicial solo se aplica durante la primera iteración de los cálculos; después se deja la red libre para que la simulación fluya de forma normal. Establecer condiciones iniciales puede ayudar en aquellas simulaciones que no funcionan correctamente (como es el caso de este ejemplo) o, incluso, en aquellas que se interrumpen, principalmente, en circuitos osciladores.



RESUMEN

Es importante saber cómo podemos enfrentar los posibles errores que se presenten al intentar simular un circuito. En este capítulo, hemos aprendido las bases del funcionamiento interno del simulador y el procedimiento para configurar sus parámetros básicos con las opciones de simulación. Además, analizamos algunos de los errores más frecuentes que suelen presentarse al tratar de simular nuestros circuitos. Por último, hemos estudiado algunas alternativas y consejos para resolverlos y, así, lograr que nuestras simulaciones siempre funcionen correctamente.

Actividades

TEST DE AUTOEVALUACIÓN

1 ¿Qué significa LCK?

2 ¿Qué es una iteración?

3 Si los cálculos se van acercando a la solución, se dice que ________.

4 ¿Cuáles son los dos métodos de integración que usa el simulador?

5 ¿Para qué sirve el anotador global?

6 ¿Qué es la anotación en tiempo real?

7 ¿Cómo se cargan los valores por defecto en las opciones de simulación?

8 ¿En dónde se configuran las opciones de simulación de forma global?

9 ¿Cómo se establece una condición inicial en una red de conexiones?

10 ¿Qué debemos escribir si necesitamos una condición inicial de 12 Volts en una red?

ACTIVIDADES PRÁCTICAS

1 Dibuje un circuito desactivando la anotación en tiempo real; al finalizarlo, use el anotador global para numerar las referencias.

2 Abra el archivo **FF.dsn**, retire la etiqueta de la condición inicial, simule el circuito y observe qué ocurre (sugerencia: espere unos 15 segundos).

3 Abra el archivo **PWM.dsn** del **Capítulo 4**, y reemplace los diodos D2 y D3 por diodos genéricos (busque la palabra diode en la ventana **Pick Devices** y elija el que dice Generic diode en la columna **Description**).

4 Una vez reemplazados los diodos en el archivo **PWM.dsn**, corra la simulación y observe qué sucede.

5 Reemplace ahora los diodos mencionados por unos 1N4001, corra la simulación y observe si el circuito funciona de forma correcta con ellos.



Servicios al lector

En esta sección presentamos un completo índice temático para encontrar en forma sencilla los conceptos fundamentales de esta obra y, además, una selección de los mejores sitios web con información, novedades y recursos relacionados con los temas que desarrollamos en este libro.

Servicio de atención al lector: usershop@redusers.com

Índice temático

E

G

H

I

L

M

Sitios web relacionados

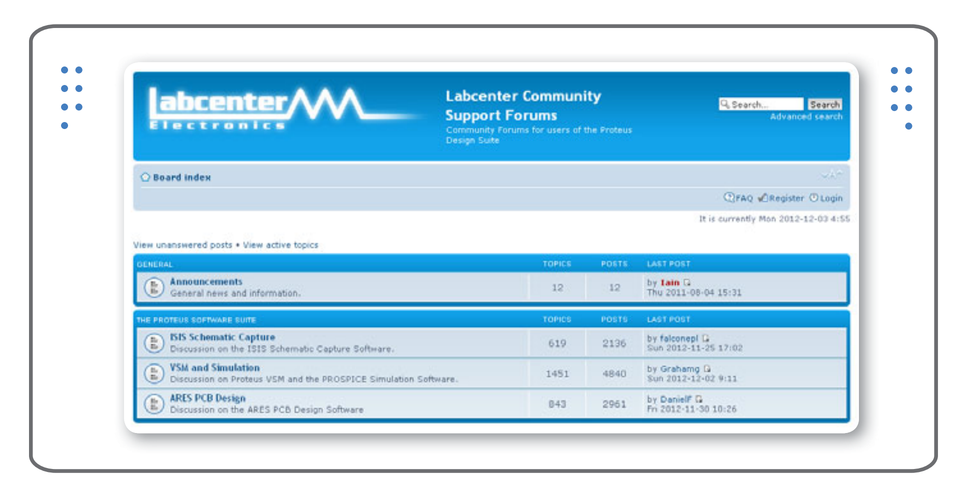
LABCENTER ELECTRONICS ● www.labcenter.com

Página web oficial del desarrollador de Proteus. Aquí podemos encontrar información relacionada con el programa, descargas, demos y complementos; y enterarnos de las novedades, últimos componentes y funciones que incluye.



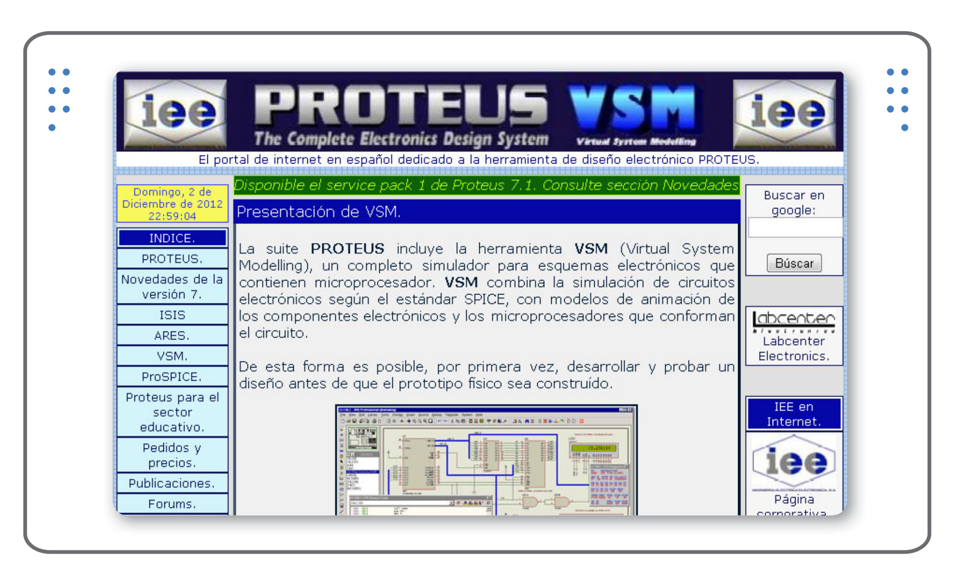
FORO OFICIAL DE SOPORTE ● http://support.labcenter.co.uk/forums

En el foro oficial de soporte de Proteus podemos registrarnos y hacer preguntas relacionadas con cualquier aspecto del programa. También es posible navegar por las consultas de otros usuarios. El sitio está principalmente en inglés.



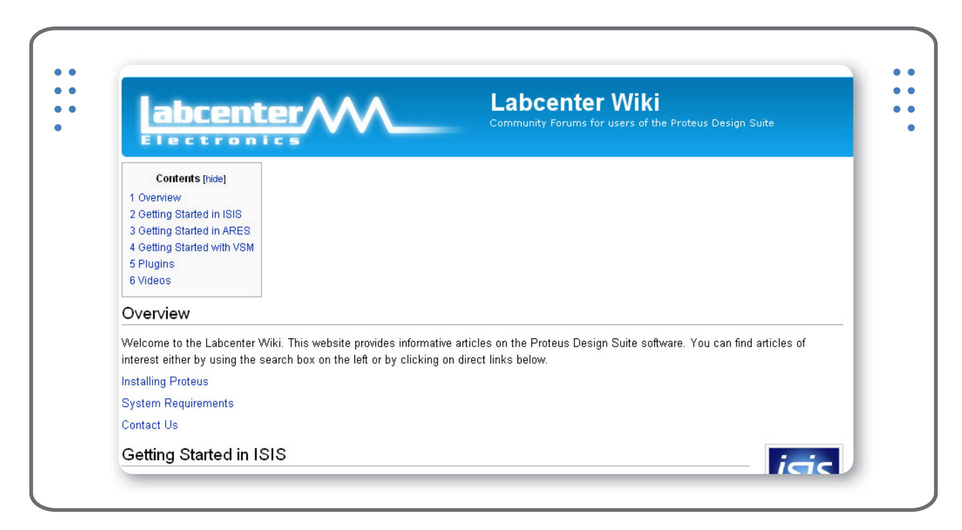
IEEPROTEUS ● www.ieespain.com/ieeproteus/vsm.html

Página web dedicada enteramente al programa Proteus, con información, tutoriales, noticias, eventos, descargas de prácticas y ejemplos, además de mucha más información relacionada con el producto. La página está en español.



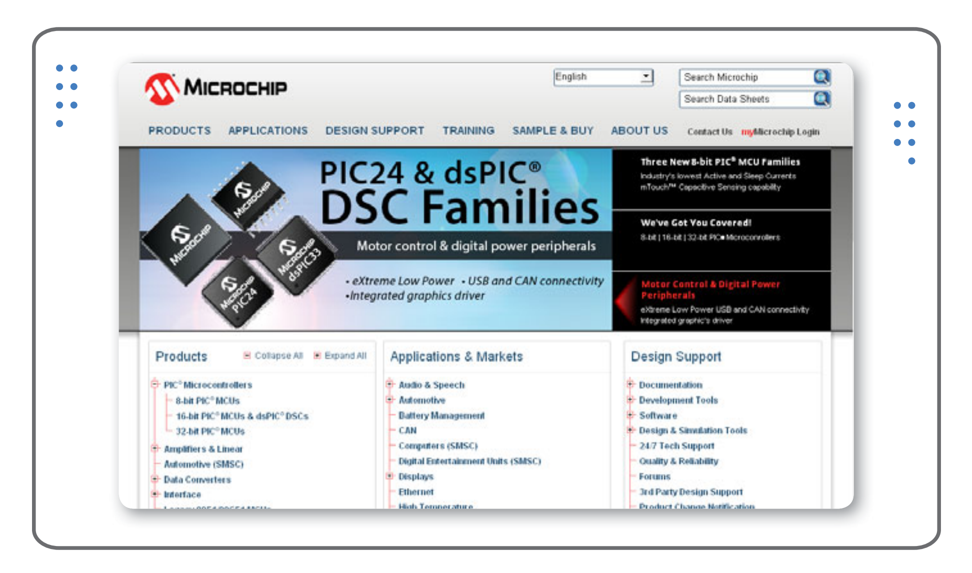
WIKI DE PROTEUS ● http://wiki.labcenter.com

Página wiki de Proteus que contiene información, guías, tutoriales y videos explicativos de muchas de sus funciones, incluyendo las del módulo ISIS y las del módulo ARES para el diseño de circuitos impresos. La página está en inglés.



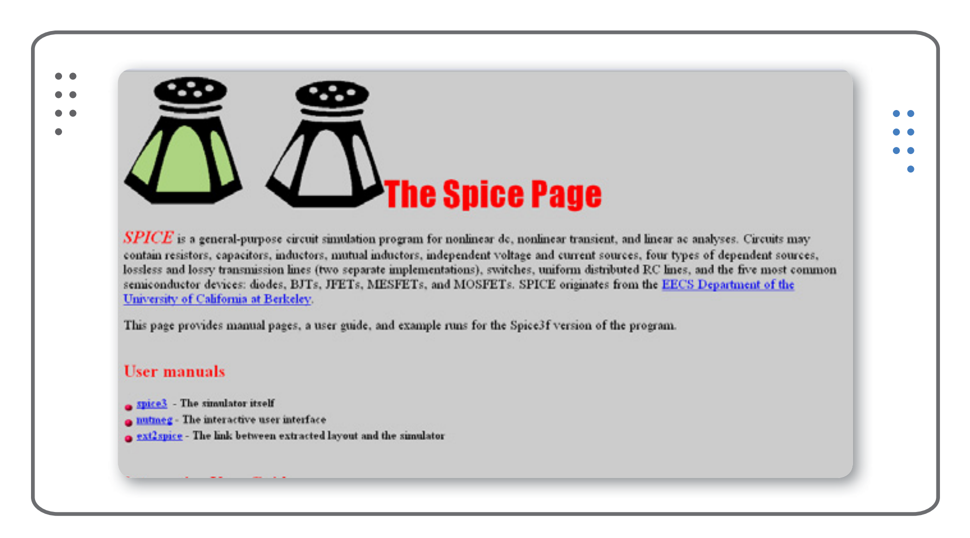
MICROCHIP ● www.microchip.com

Sitio oficial del fabricante de microcontroladores PIC. Desde aquí podemos descargar el entorno de desarrollo MPLAB IDE para escribir y ensamblar los programas para PICs, en el cual se puede integrar Proteus.



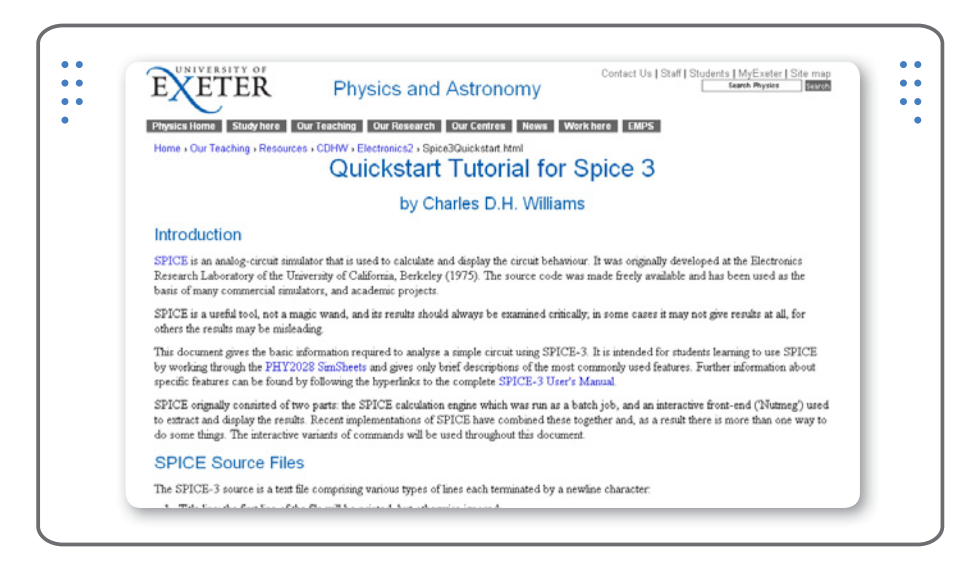
SPICE ● http://bwrc.eecs.berkeley.edu/Classes/IcBook/SPICE

Página oficial del motor de simulación SPICE, en donde encontraremos información, tutoriales y ejemplos del uso de este simulador, que es el corazón de Proteus. Esta página se encuentra en inglés.



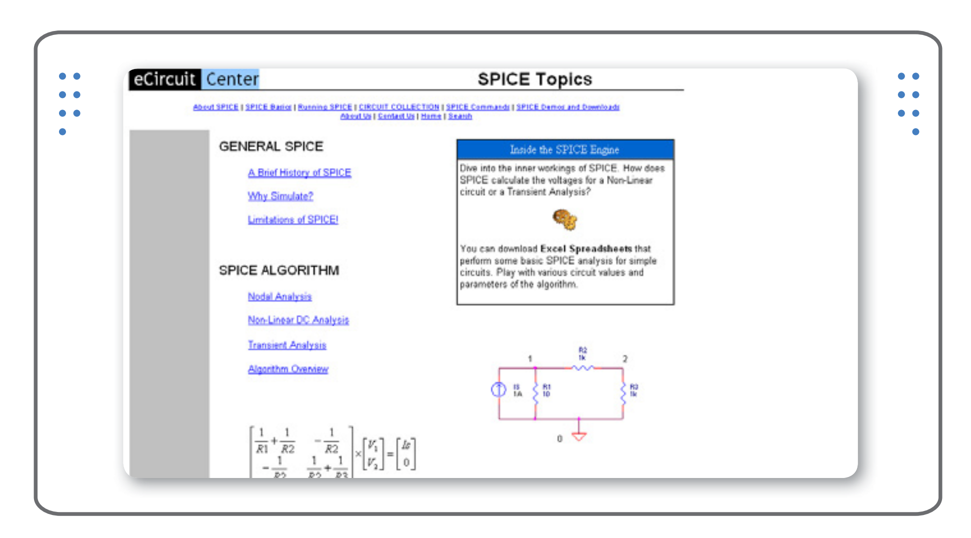
GUÍA SPICE ● http://newton.ex.ac.uk/teaching/CDHW/Electronics2/

Si vamos al link **Electronics Learning-Resources on the WWW**, y buscamos la sección **SPICE**, accedemos a una guía rápida sobre el uso de este motor de simulación, dedicada a los interesados en aprender su funcionamiento interno.



SPICE TOPICS ● www.ecircuitcenter.com/SPICEtopics.htm

En este sitio encontramos información sobre SPICE, entre la cual se destaca una breve historia del nacimiento y el desarrollo de este motor de simulación, además de otros temas de interés, como, por ejemplo, sus limitaciones. Está en inglés.



DATASHEET CATALOG ● www.datasheetcatalog.com

Siempre es útil contar con hojas de datos para ver el orden, la función o la configuración de los pines de los componentes. En este sitio hay una completa colección de hojas de datos de todo tipo de componentes de manera gratuita.



FOROS DE ELECTRÓNICA ● www.forosdeelectronica.com

Este es el foro más importante en idioma español sobre temas relacionados con la electrónica, en donde los usuarios realizan y responden todo tipo de preguntas. Incluye una sección dedicada a resolver dudas de programas de simulación.



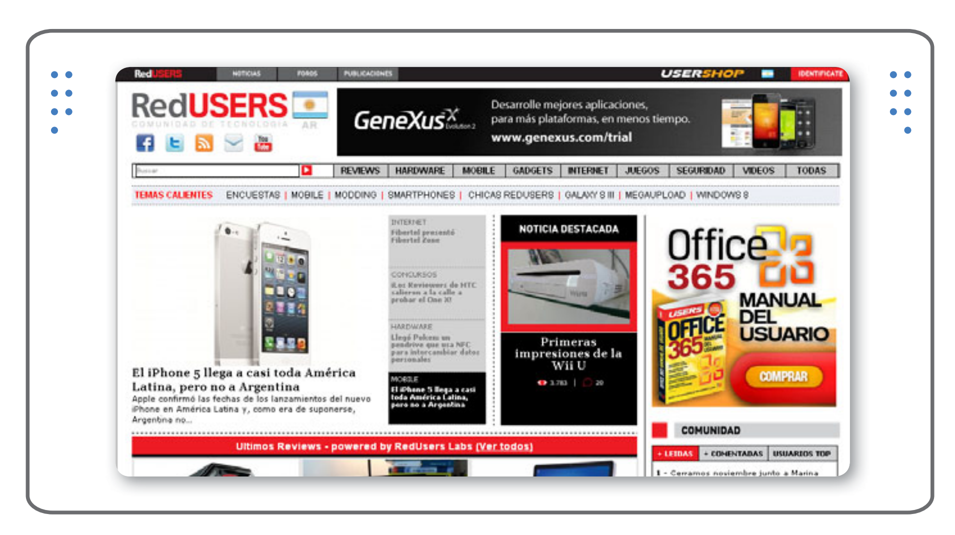
MODELOS PROTEUS ● www.sonelec-musique.com/logiciels.html

Desde la sección **Ressources pour CAO Proteus**, podemos descargar, de manera gratuita, modelos de componentes y vistas en 3D para el módulo ARES. Contiene, además, un tutorial para crear vistas en 3D. La página está en inglés y francés.



REDUSERS ● www.redusers.com/premium/notas_contenidos/

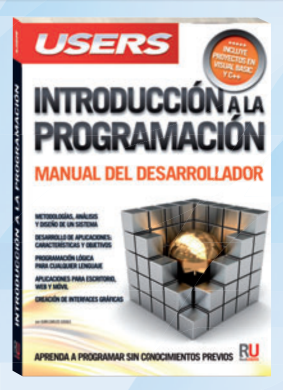
Este es el sitio de la editorial RedUsers. Desde la sección **Publicaciones/Libros**, podemos descargar los archivos de ejemplos y proyectos utilizados a lo largo de este libro, para tenerlos en nuestra PC y usarlos cada vez que sea necesario.





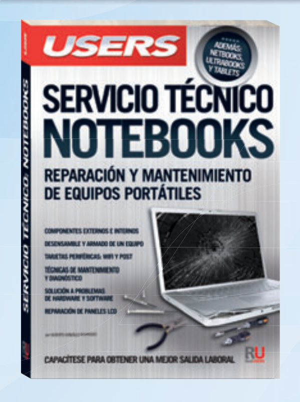
Esta obra reúne todos los conocimientos teóricos y prácticos para conver¬tirse en un técnico especializado en Windows.

→ 320 páginas / ISBN 978-987-1857-70-8



Un libro ideal para iniciarse en el mundo de la programación y conocer las bases necesarias para generar su primer software.

→ 384 páginas / ISBN 978-987-1857-69-2



Presentamos una obra fundamental para aprender sobre la arquitectura física y el funcionamiento de los equipos portátiles.

→ 352 páginas / ISBN 978-987-1857-68-5



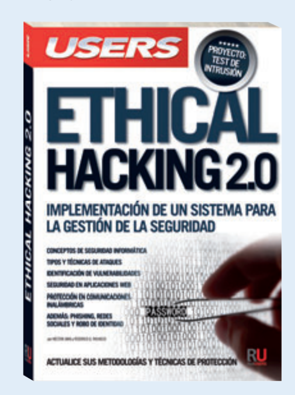
Una obra ideal para aprender todas las ventajas y servicios integrados que ofrece Office 365 para optimizar nuestro trabajo.

→ 320 páginas / ISBN 978-987-1857-65-4



Esta obra presenta las mejores aplicaciones y servicios en línea para aprovechar al máximo su PC y dispositivos multimedia.

→ 320 páginas / ISBN 978-987-1857-61-6



Esta obra va dirigida a todos aquellos que quieran conocer o profundizar sobre las técnicas y herramientas de los hackers.

→ 320 páginas / ISBN 978-987-1857-63-0



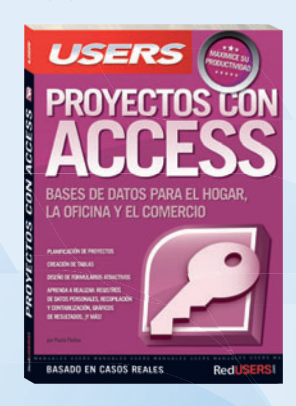
Este libro se dirige a fotógrafos amateurs, aficionados y a todos aquellos que quieran perfeccionarse en la fotografía digital.

→ 320 páginas / ISBN 978-987-1857-48-7



En este libro encontraremos una completa guía aplicada a la instalación y configuración de redes pequeñas y medianas.

→ 320 páginas / ISBN 978-987-1857-46-3



Esta obra está dirigida a todos aquellos que buscan ampliar sus conocimientos sobre Access mediante la práctica cotidiana.

→ 320 páginas / ISBN 978-987-1857-45-0

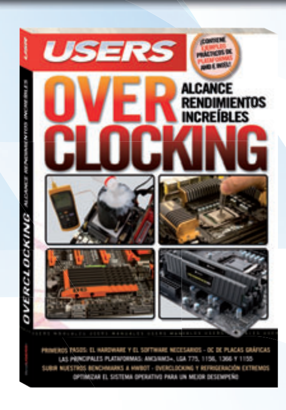
Este libro nos introduce en el apasionante mundo del diseño y desarrollo web con Flash y AS3.

320 páginas / ISBN 978-987-1857-40-1



Esta obra presenta un completo recorrido a través de los principales conceptos sobre las TICs y su aplicación en la actividad diaria.

320 páginas / ISBN 978-987-1857-41-8



Este libro está dirigido tanto a los que se inician con el overclocking, como a aquellos que buscan ampliar sus experiencias.

320 páginas / ISBN 978-987-1857-30-2



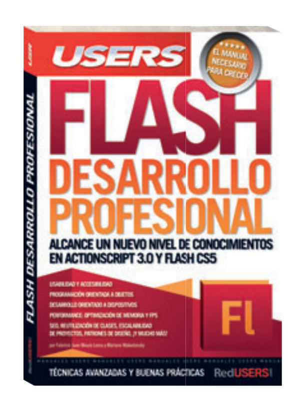
Este manual único nos introduce en el fascinante y complejo mundo de las redes inalámbricas.

320 páginas / ISBN 978-987-1773-98-5



Esta increíble obra está dirigida a los entusiastas de la tecnología que quieran aprender los mejores trucos de los expertos.

320 páginas / ISBN 978-987-1857-01-2



Esta obra se encuentra destinada a todos los desarrolladores que necesitan avanzar en el uso de la plataforma Adobe Flash.

320 páginas / ISBN 978-987-1857-00-5



Un libro clave para adquirir las herramientas y técnicas necesarias para crear un sitio sin conocimientos previos.

320 páginas / ISBN 978-987-1773-99-2



Una obra para aprender a programar en Java y así insertarse en el creciente mercado laboral del desarrollo de software.

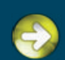
352 páginas / ISBN 978-987-1773-97-8



Este libro presenta un nuevo recorrido por el máximo nivel de C# con el objetivo de lograr un desarrollo más eficiente.

320 páginas / ISBN 978-987-1773-96-1

Descargue un capítulo gratuito
Entérese de novedades y lanzamientos



Compre los libros desde su casa y con descuentos



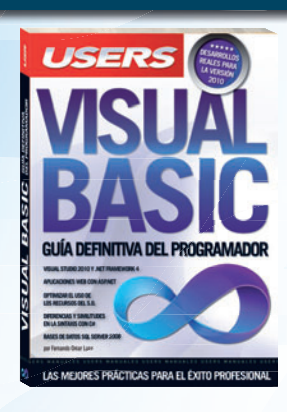
Esta obra presenta todos los fundamentos y las prácticas necesarios para montar redes en pequeñas y medianas empresas.

→ 320 páginas / ISBN 978-987-1773-80-0



Una obra única para aprender sobre el nuevo estándar y cómo aplicarlo a nuestros proyectos.

→ 320 páginas / ISBN 978-987-1773-79-4



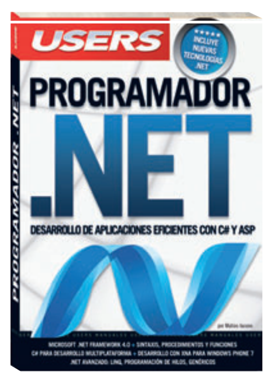
Un libro imprescindible para aprender cómo programar en VB.NET y así lograr el éxito profesional.

→ 352 páginas / ISBN 978-987-1773-57-2



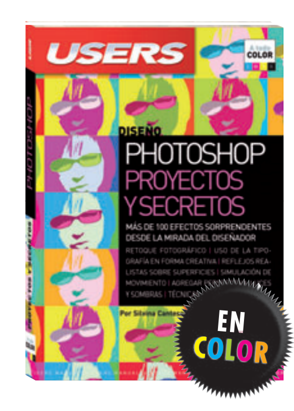
Una obra para aprender los fundamentos de los microcontroladores y llevar adelante proyectos propios.

→ 320 páginas / ISBN 978-987-1773-56-5



Un manual único para aprender a desarrollar aplicaciones de escritorio y para la Web con la última versión de C#.

→ 352 páginas / ISBN 978-987-1773-26-8



Un manual imperdible para aprender a utilizar Photoshop desde la teoría hasta las técnicas avanzadas.

→ 320 páginas / ISBN 978-987-1773-25-1



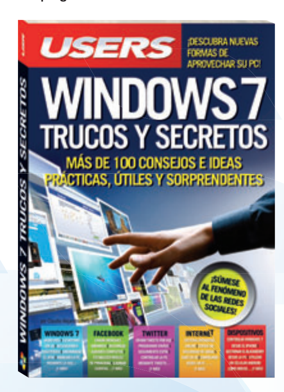
Una obra imprescindible para quienes quieran conseguir un nuevo nivel de profesionalismo en sus blogs.

→ 352 páginas / ISBN 978-987-1773-18-3



Un libro único para ingresar en el apasionante mundo de la administración y virtualización de servidores.

→ 352 páginas / ISBN 978-987-1773-19-0



Esta obra permite sacar el máximo provecho de Windows 7, las redes sociales y los dispositivos ultraportátiles del momento.

→ 352 páginas / ISBN 978-987-1773-17-6

 + 54 (011) 4110-8700  usershop@redusers.com

Descargue un capítulo gratuito
Entérese de novedades y lanzamientos

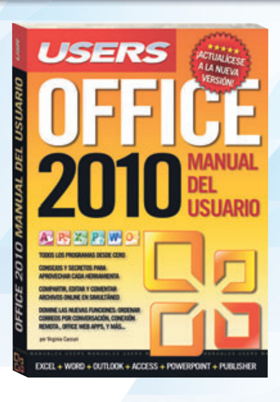


Compre los libros desde su casa
y con descuentos



Un libro fundamental para aprender a trabajar de forma profesional con las herramientas audiovisuales de Adobe.

→ 320 páginas / ISBN 978-987-26013-9-3



Un manual ideal para aprender todo sobre la nueva versión de Office y las posibilidades de trabajo online que ofrece.

→ 352 páginas / ISBN 978-987-26013-6-2



Un manual imperdible para aprender a usar este programa y aprovechar todas sus posibilidades al máximo.

→ 320 páginas / ISBN 978-987-26013-4-8



Esta obra única nos introduce en .NET para aprender sobre la última versión del lenguaje más utilizado de la actualidad.

→ 400 páginas / ISBN 978-987-26013-5-5



Este libro imprescindible nos enseña cómo mantener nuestra información protegida de todas las amenazas de la Web.

→ 320 páginas / ISBN 978-987-26013-1-7



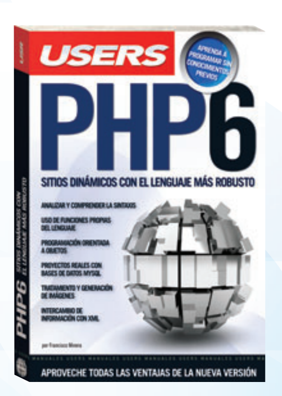
Un manual imperdible para guardar como guía de referencia y para aplicar siempre ante entornos complejos.

→ 368 páginas / ISBN 978-987-26013-0-0



Un libro imprescindible para exprimir al máximo las capacidades multimedia que ofrecen Internet y Windows 7.

→ 352 páginas / ISBN 978-987-663-036-8



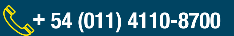
Una obra fundamental para aprender a programar desde cero con la última versión del lenguaje más robusto.

→ 368 páginas / ISBN 978-987-663-039-9



Este libro único nos brindará todas las respuestas para dominar los dos blogs más populares de la Web: Blogger y WordPress.

→ 320 páginas / ISBN 978-987-663-037-5



+ 54 (011) 4110-8700 usershop@redusers.com



PROTEUS VSM



El diseño asistido por computadora es cada vez más utilizado en todas las ramas de la ingeniería, y la electrónica encuentra en Proteus VSM un poderoso aliado en el diseño y el análisis de circuitos.

Esta obra fundamental está dirigida a todos aquellos aficionados y expertos en el área que deseen aprender a fondo sobre la simulación de circuitos electrónicos en Proteus VSM. A lo largo de sus páginas, haremos un recorrido por sus herramientas virtuales, desde la elección y el manejo de componentes, hasta las correcciones de los errores más frecuentes en simulación.

Al completar la lectura de este libro, el lector conocerá este poderoso laboratorio virtual que le permitirá acelerar y simplificar los procesos de diseño y prueba de circuitos electrónicos, sin necesidad de construirlos físicamente.



Trabajar con Proteus es como hacerlo en la realidad, pero con componentes interactivos que nos permiten ahorrar tiempo y dinero.



* EN ESTE LIBRO APRENDERÁ:

- **Introducción:** conceptos básicos sobre Proteus y la simulación en ISIS. Qué componentes conviene utilizar y cuáles son sus ventajas principales.
- **Señales y medición:** diferentes formas de generar señales análogas y digitales en Proteus. Cuáles son los instrumentos de medición y cómo se utilizan.
- **Análisis con gráficos:** creación y evaluación de los diferentes tipos de gráficos: analógicos digitales, mixtos e interactivos, entre otros.
- **Microcontroladores:** simulación de circuitos, depuración del código fuente y operación sin necesidad de un ensamblador externo.
- **Diagramas y componentes:** dibujo de diseños en múltiples hojas y uso de subcircuitos. Cómo exportar listas de componentes y crear nuevos símbolos desde cero.
- **Interfaz y correcciones:** personalización de las interfaces en función de las necesidades y consejos para corregir los errores más frecuentes en simulación.



» SOBRE EL AUTOR

Víctor Rossano es Ingeniero en Electrónica con especialidad en Sistemas Digitales. Cursó la carrera en la Escuela Superior de Ingeniería Mecánica y Eléctrica, en el Instituto Politécnico Nacional de la ciudad de México. Apasionado por la ciencia y la tecnología, ya ha escrito dos libros para esta misma editorial.

» NIVEL DE USUARIO

Intermedio / Avanzado

» CATEGORÍA

Electrónica / Hardware

